

**VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky**

**Otřesové čidlo a jeho komunikace v konceptu IoT
Shock Sensor and its Communication in IoT Concept**

2017

Erik Grešák

Zadání diplomové práce

Student: **Bc. Erik Grešák**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T059 Mobilní technologie

Téma: **Otřesové čidlo a jeho komunikace v konceptu IoT**
Shock Sensor and its Communication in IoT Concept

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem diplomové práce je návrh a praktická realizace otřesového čidla a IoT infrastruktury dle specifikace LoRaWAN, která bude kromě čidla obsahovat IoT bránu a serverovou část s přístupem uživatele k údajům senzoru.

1. Představení konceptu Internet věcí a senzorových sítí.
2. Otevřený standard LoRaWAN.
3. Návrh LoRaWAN infrastruktury a její praktická realizace.
4. Návrh otřesového čidla a jeho integrace do LoRaWAN infrastruktury.
5. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:


1. N. Sornin, M. Luis, T. Eirich and T. Kramp, Release V1.0 Reference implementation and documentation of a LoRa network node. Last access March 2016, URL <https://github.com/Lora-net>
2. S. Monk, Programming Arduino. McGraw-Hill, 2012, ISBN 9780071784221.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí diplomové práce: **doc. Ing. Miroslav Vozňák, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017


doc. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry





prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou/diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 28. dubna 2017


.....
podpis studenta

Poděkování

Za ochotu a čas mně věnovaný děkuji svému vedoucímu doc. Ing. Miroslavu Vozňákovi, Ph.D., který byl v roli mentora mou oporou během řešení této diplomové práce. V neposlední řadě poděkování patří i kolegům a kamarádům z komunity LIPTEL působící na FEI VŠB-TUO, které jsem součástí, a kde jsem našel během svého magisterského studia zázemí, a to nejen v rovině odborné.

Prohlášení zástupce spolupracující právnické nebo fyzické osoby

„Souhlasím se zveřejněním této bakalářské/diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-TU Ostrava.“

Dne: 28. dubna 2017

..........
podpis zástupce

Abstrakt

Diplomová práce se zabývá návrhem a realizací otřesového čidla s možností jeho integrace do IoT infrastruktury. Pro tuto integraci je vytvořena infrastruktura dle specifikace LoRa WAN, která kromě samotného čidla obsahuje bránu a serverovou část s přístupem uživatelů do webové aplikace. Aplikace umožňuje uživateli zobrazit získané data v textové podobě, nebo pomocí podaplikace, dále umožňuje spravovat bránu a další čidla. Praktická realizace brány a severu jsou rovněž klíčovými body této práce, kde hrál významnou roli MQTT protokol spolu s LoRa technologií. Pro měření otřesu je použit pasivní otřesový prvek, jehož stav je kontrolován osmi bitovým mikroprocesorem Atmega328, data informující o změně stavu jsou odeslána vysílačem RFM96W na bránu. Čidlo je provozováno na bateriovém článku, a proto je v návrhu dbáno na minimalizaci odběru elektrické energie. Krabice pro finální návrh otřesového čidla byla vytištěna na 3D tiskárně. V současnosti je vytvořená infrastruktura v rámci této práce provozována v kampusu VŠB-TUO, kde probíhá sběr dat z různých čidel.

Klíčová slova

IoT, LoRa WAN, MQTT, Atmega, Arduino, Infrastruktura, Webová aplikace, Brána, Server, Otřesové čidlo, Senzor

Abstract

The diploma thesis deals with a design and implementation of a vibration sensor providing the possibility of integration in the Internet of things (IoT) infrastructure. An infrastructure in accordance with the LoRa WAN specification has been created for this integration, which, apart from the sensor, also comprises gate and server with users' access to the web application. The application enables users to display acquired data as text, or to use a sub-application, and also enables administration of the gate as well as other sensors. The creation of the gate and server are the key points of the present paper, in which MQTT protocol and LoRa technology played a major role. Vibrations are measured using a passive vibration element, with Atmega328 8-bit microprocessor checking its state; data about the state are sent by the RFM96W transmitter to the gate. The sensor is placed on a battery, therefore the design considers minimum electricity consumption. The solution also applies a 3D printer, onto which the box for the vibration sensor is fitted. Currently, the whole infrastructure is being tested on campus, where data is being gathered from several sensors.

Key words

IoT, LoRa WAN, MQTT, Atmega, Arduino, Infrastructure, Web application, Gateway, Server, Shock sensor, Sensor

Obsah

Seznam použitých symbolů.....	- 11 -
Seznam ilustrací a seznam tabulek.....	- 12 -
Úvod.....	- 14 -
1 IoT, Internet věcí.....	- 15 -
1.1 Faktory podporující rozvoj internetu věcí.....	- 15 -
1.2 Překážky internetu věcí.....	- 15 -
2 LoRa, Long Range	- 16 -
2.1 LoRa WAN	- 17 -
2.1.1 Třídy	- 17 -
3 Přehled výrobců LoRa modulů	- 18 -
3.1 Přehled výrobců modulů	- 18 -
3.1.1 Semtech	- 18 -
3.1.2 HOPE Microelectronics.....	- 18 -
3.1.3 Laird	- 18 -
3.1.4 Microchip	- 18 -
3.1.5 IMTS GmbH.....	- 18 -
3.2 Rozbor modulů.....	- 19 -
3.2.1 Rádiový modul RFM95W	- 19 -
3.2.2 Rádiový modul Semtech SX1276	- 19 -
3.2.3 Rádiový modul Laird RM1xx Series.....	- 20 -
3.2.4 Rádiový modul Microchip RN2483	- 20 -
3.2.5 iU880B - Long Range USB Adapter.....	- 20 -
3.2.6 iC880A - LoRa WAN Koncentrátor.....	- 20 -
4 Gateway.....	- 21 -
4.1 Raspberry Pi.....	- 21 -
4.1.1 GPIO Rozhraní	- 22 -
4.2 Komunikace	- 22 -
4.3 Konstrukce brány	- 25 -
5 Server	- 27 -
5.1 Komunikace	- 27 -
5.1.1 ACL	- 27 -

5.1.2	MQTT.....	- 27 -
5.2	Databáze MySQL.....	- 28 -
5.3	LAMP	- 30 -
6	Webová aplikace	- 31 -
6.1	Technologie pro realizaci webové aplikace	- 31 -
6.1.1	HTML.....	- 31 -
6.1.2	CSS.....	- 32 -
6.1.3	XML	- 32 -
6.1.4	PHP.....	- 33 -
6.1.5	JavaScript	- 33 -
6.2	Mapy	- 33 -
6.3	Správa brán a zařízení.....	- 34 -
6.4	Aplikace senzorů.....	- 36 -
7	Rozdělení senzorů	- 37 -
7.1	Aktivní senzory	- 37 -
7.2	Pasivní senzory	- 37 -
7.3	Čidla snímání otřesů.....	- 37 -
7.3.1	Rtuťové čidlo.....	- 37 -
7.3.2	Kuličkové čidlo	- 38 -
7.3.3	Pružinové čidlo.....	- 38 -
8	Návrh otřesového čidla.....	- 39 -
8.1	Mikroprocesor.....	- 39 -
8.1.1	Požadavky	- 39 -
8.1.2	Mikroprocesor Atmega328.....	- 39 -
8.1.3	Programování mikroprocesoru	- 39 -
8.2	Návrh plošného spoje, DPS.....	- 40 -
9	Realizace otřesového čidla	- 42 -
9.1	Struktura kódu.....	- 42 -
9.2	Výroba plošného spoje, DPS.....	- 43 -
10	Návrh krabičky.....	- 45 -
10.1	3D Tiskárny.....	- 45 -
10.1.1	Technologické rozdělení	- 45 -
10.2	Použitá tiskárna	- 47 -

10.3 Realizace	- 48 -
11 VŠB LoRa - Testbed	- 49 -
Závěr	- 51 -
Použitá literatura	- 52 -
A Příloha	- 55 -

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
U	V	Napětí
Ah	Ah	Ampérhodina
I	A	Proud
Hz	Hz	Hertz
B	B	Bajt
SNR	-	Poměr signál-šum
RSSI	dBm	Decibel miliwatt
DATA SIZE	B	Bajt

Seznam ilustrací a seznam tabulek

Číslo ilustrace	Název ilustrace	Číslo stránky
2.1	Diagram autentizace metody OTAA	17
3.1	Dragino shield pro Raspberry Pi	19
4.1	Raspberry Pi 3, model B	22
4.2	MQTT komunikace využívající bridge	24
4.3	MQTT komunikace Server a Klient	24
4.4	Montážní krabice pro bránu.	25
4.5	Regulátor napětí, bleskojistka a SAW filtr.	26
5.1	Relační model databáze	29
6.1	Zobrazení senzoru v mapě.	33
6.2	Správa brány	33
6.3	Správa zařízení	34
6.4	Graf informující o signálu v době přijetí zprávy	35
6.5	Zobrazení dat v aplikaci Temperature	35
7.1	Olověné čidlo	36
7.2	Kuličkové čidlo	38
7.3	Pružinové čidlo	38
8.1	SPI	40
8.2	Blokové schéma čidla	41
8.3	Deska plošného spoje a rozložení součástek	41
9.1	Výsledný DPS	43
10.1	Selective laser sintering	45
10.2	Fused deposition modeling	46
10.3	Stereilitography	46
10.4	Eletronic beam melting	47
10.5	Návrh krabičky	48
10.6	Sestavené čidlo spolu s krabičkou	48
11.1	Koncept komunikace	49
11.2	Pokrytí testbedu	50

Seznam ilustrací a seznam tabulek

Číslo tabulky	Název tabulky	Číslo stránky
2.1	Kmitočtová pásma a kanály	16
4.1	Operační systémy	21

Úvod

V současné době je Internet věcí (IoT) fenoménem, ačkoliv myšlenka je starší a diskutována již přes deset let, tak v posledních letech s příchodem tématu Průmyslu 4.0 se oblast IoT začala dynamicky rozvíjet. A to díky svým možnostem využití, které nalezneme v oblasti průmyslu a různých chytrých řešení, jejichž smysl je především v rovině usnadnění života formou zautomatizování různých činností, což má nesporný přínos pro společnost. V domácnosti nám může přinést vysoký komfort a v průmyslu ušetřit nemalé prostředky. Základním principem je přenos informace vyslané ze senzoru do sítě internetu. Existují různá řešení infrastruktur založená na rozdílných technologiích zprostředkování komunikace. Jedním ze standardů je LoRa infrastruktura, která je díky své oblibě čím dál tím více začleněná do IoT prostředí, kde je pevně stanoven komunikační protokol LoRa WAN. Díky svému potenciálu a využití nepoplatných frekvenčních pásem je správnou volbou pro realizaci nízko nákladové sensoriky. Jedním z takovýchto senzorů může být otřesové čidlo, které při otřesu indikuje událost. Takové čidlo je možné uplatnit ve spoustě aplikacích, například pro zabezpečení majetku proti krádeži, indikaci spadlého stromu nebo zabránění vandalismu na veřejném dopravním značení. Cílem této diplomové práce je návrh a praktická realizace otřesového čidla a IoT infrastruktury dle specifikace LoRa WAN, která bude kromě čidla obsahovat IoT bránu a serverovou část s přístupem k údajům senzoru. Následující práce je rozdělená do jedenácti kapitol. V první části práce je vysvětleno co je to IoT, dále je představen koncept senzorových sítí a jejich překážky či faktory rozvoje. V další kapitole je popsána samotná technologie LoRa včetně LoRa WAN standardu. V popisu nalezneme specifikace určující hodnoty dosahu, frekvenční pásma a třídy komunikace. V třetí kapitole je provedena rešerše dostupných modulů pro uvedenou infrastrukturu. Rešerše dále obsahuje informace a zaměření firem, vyrábějící uvedené moduly. Výstupem je výběr vhodného modulu pro následné řešení infrastruktury. Následující kapitoly se zabývají gateway (brána) a serverem. Jsou zde popsány technologie a jednotlivé komponenty, které byly využity pro realizaci brány a serveru, včetně popisu rozhraní a komunikace. Dále je popsána samotná konstrukce brány a instalace konstrukčních prvků s možností instalace a provozu brány na venkovním prostředí. Z uživatelského hlediska, aby koncový uživatel měl přehled o svých senzorech a datech, které jsou těmito senzory získané, se část práce věnuje vývojem webové aplikace uvedené v šesté kapitole. Jsou zde popsány základní prvky, spolu s rozhraním umožňující správu zařízení a brán zaregistrovaných v aplikaci. V dnešní době existuje řada různých senzorů a čidel komunikující v senzorové síti. Otřesová čidla jsou poměrně známá i v běžném životě, například v automobilech jako spínací prvek pro alarm, nebo detekce autonehody. Součástí práce je realizace a návrh otřesového čidla, kde bylo provedeno základní rozdělení senzorů a technický rozbor pasivních typů otřesových detektorů. Následně z poznatků a vhodného výběru prvku pro detekci otřesu, byl sestaven návrh, pomocí využití editoru EAGLE pro vytváření předlohy na DPS. Je zde také uveden popis a postup výroby samotného plošného spoje, včetně osazení součástek, struktury kódu a implementace knihoven. Před realizací návrhu je zde vybrán mikroprocesor s ohledem na požadavky. Celá sestava byla při návrhu testována na zvolené vývojové platformě Arduino, pomocí které byl také samotný program uveden do mikroprocesoru. Předposlední kapitola je určena pro návrh krabičky v modelovacím nástroji a její tisk pomocí 3D tiskárny. Zároveň jsou zde rozebrány různé technologie a metody tisku pomocí 3D tiskáren. V poslední kapitole je uvedena implementace infrastruktury v podobě vytvoření testovacího testbedu v Porubském kampusu Vysoké školy báňské - Technické univerzity Ostrava. V závěru jsou pak zhodnoceny výsledky práce a testovacího prostředí vybudovaného na porubském kampusu.

1 IoT, Internet věcí

Jedná se nový trend v oblasti kontroly a komunikace předmětů běžného využití a to zejména prostřednictvím technologií bezdrátového přenosu dat a internetu. Takto propojená zařízení umožní sběr velkého množství dat, která lze dále zpracovávat a využívat v nejrůznějších oblastech jako logistika, záchranné složky, energetika, doprava, meteorologie atd. Již dnes v praxi funguje nespočet zařízení, jako dálkově ovládané spotřebiče, zásuvky, osvětlení, kamery, meteostanice či jiné dostupné senzory [13]. Prozatím však nespolečně spolupracují pod jednou technologií a společným protokolem. V současné době je internetem propojeno okolo 14 miliard objektů. V roce 2020 jich může být podle zprávy okolo 20 až 100 miliard. Odhadované potenciální úspory činí kolem 6,2 bilionu dolarů [22]. Již nyní existuje více připojených objektů, než je lidí na planetě. Senzorové sítě WSN (Wireless sensor networks) je možné obecně popsat jako síť uzlů, které společně komunikují a řídí prostředí. Tím je docíleno interakce mezi osobami nebo počítači a okolním prostředím [10]. Dnes jsou senzorové sítě složeny převážně z uzlů, které představují čidla, brány a klienty. Velké množství nasazených uzlů v blízkosti monitorovací oblasti, vytvářejí síť prostřednictvím vlastní organizace. Uzly senzorů sledují nashromážděná data a následně je přenášejí dál skrze další čidla. Během zpracování zprávy může být informace přeposlána mezi několika uzly čidel až ke koncovému uzlu brány a předány do Internetu. Náklady na WSN dramaticky klesly a využití se postupně rozšiřuje do průmyslových a komerčních oblastí. Jeden z příkladů spadající pod standard pro technologie WSN je dobře známé Zigbee [14][18].

1.1 Faktory podporující rozvoj internetu věcí

Miniaturizace součástek a zmenšení polovodičů má za následek minimalizaci ceny. Tento faktor nám tak umožňuje pořídit senzor v řádech několika dolarů. Dalším faktorem je sdílení výpočetních prostředků skrze cloud řešení a snížení tak ceny za provoz. IoT cloud [12] je platforma, která je určena pro ukládání a zpracování dat z internetu věcí. Tyto platformy jsou uzpůsobené tak, aby dokázaly zpracovat příliv masivního objemu dat generovaných pomocí senzorů připojených do sítě. Dále dotazování webových stránek a různých aplikací včetně interaktivního uživatelského prostředí s real-time odezvou. Pomocí cloudu je možné přijímat miliardy různých událostí za jediný den, které je možné využít pro velkokapacitní analytiku. Zpracování datové analýzy probíhá v téměř reálném čase a umožňují nám tak napomáhat například pro burzovní operace, dopravní služby, ale i přizpůsobení chování větrné elektrárny na základě aktuálních údajů o počasí. Cloud řešení může zahrnovat aplikační logiku, vizualizaci, analýzu, řízení procesu, úložiště dat zařízení, registraci a identifikaci, správu API, transformaci a propojení.

1.2 Překážky internetu věcí

Každá větší skupina technologického odvětví řeší problémy a není tomu jinak ani u IoT. Absence neexistujícího stabilního obchodního modelu, bez kterého vzniká nejistota zisku na straně dodavatelů. Náročnost na technologie pro zpracování velkého množství dat. Rychlý rozvoj a existence nových firem, jejichž plány mohou změnit původní obchodní model investorů. Nestabilní standardy a vytváření norem vyhovující silným hráčům. Zabezpečení a důvěra, kde velké množství připojených zařízení bude vystaveno bezpečnostním hrozbám. Hrozby zneužití dat, například aktuální spotřeba energie, nebo obyčejný teploměr může prozradit, zda se někdo v objektu nachází či nikoliv.

2 LoRa, Long Range

Technologie je určena pro internet věcí IoT. Umožňuje instalaci na velké rozloze s malým datovým tokem. Tato technologie je patentována firmou Semtech [25] pod protokolem LoRa WAN, který zajišťuje transparentní zabezpečený přenos dat mezi koncovým zařízením a aplikací běžící na serveru a naopak. O standardizaci a rozvoj se stará organizace LoRa Alliance tvořící společenstvo firem. LoRa je navržena jak pro evropské pásmo 868 Mhz, 433 Mhz, tak pro americké 913Mhz s citlivostí až -136 dB a s odolností vůči rušení -16 dB pod úrovní šumu. Disponuje dosahem na přímou viditelnost až okolo 40 km. Ve městských zástavbách je tato vzdálenost menší a to díky překážkám v cestě signálu představující budovy, maximální vzdálenost se tak zmenší až na 2 km. Modulace LoRa je patentově chráněný přístup, založený na modulaci Chirp [24].

Tabulka 2.1: *Kmitočtová pásma a kanály [33]*

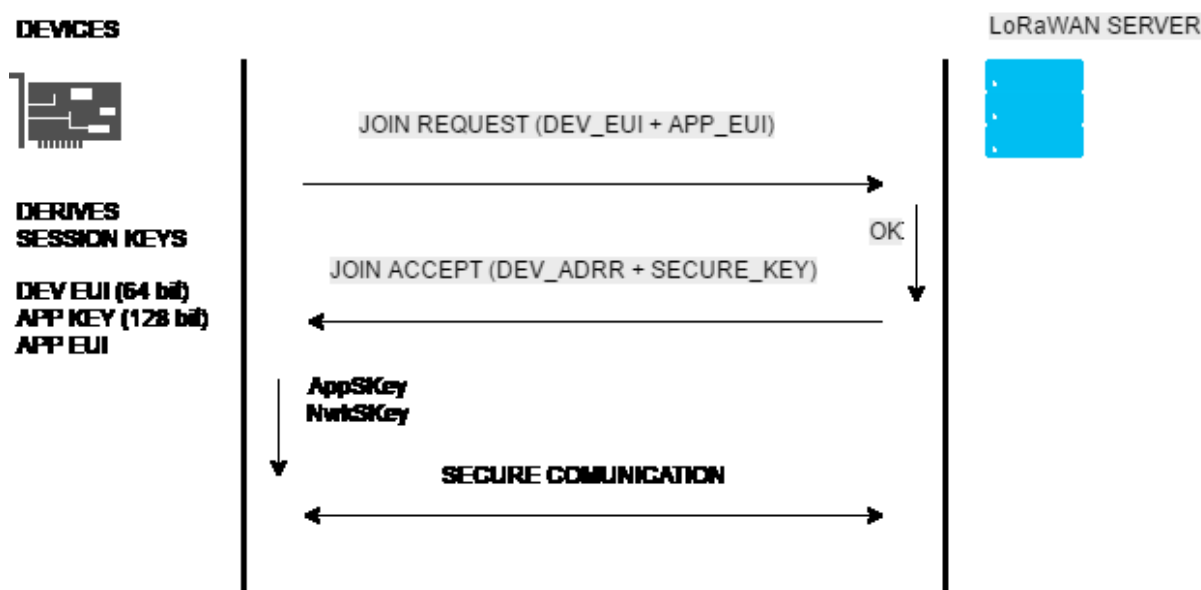
Region	LoRa frequency band	LoRa channel frequency
EU	863 to 870 MHz	868.10 Mhz (used by Gateway to listen) 868.30 MHz (used by Gateway to listen) 868.50 MHz (used by Gateway to listen) 864.10 MHz (used by End device to transmit Join Request) 864.30 MHz (used by End device to transmit Join Request) 864.50 MHz (used by End device to transmit Join Request) 868.10 MHz (used by End device to transmit Join Request) 868.30 MHz (used by End device to transmit Join Request) 868.50 MHz (used by End device to transmit Join Request)
US	902 to 928 MHz	902.3 MHz to 914.9 MHz spaced at 200KHz (Upstream-64 channels) 903 MHz to 914.2 MHz spaced at 1.6 MHz apart (Upstream- 8 channels) 923.3 MHz to 927.5 MHz spaced at 600KHz apart (Downstream- 8 channels)
China	779 to 787 MHz	779.5 MHz (Default channel) 779.7 MHz (Default channel) 779.9 MHz (Default channel) 779.5 MHz (Used by ED to transmit Join Request) 779.7 MHz (Used by ED to transmit Join Request) 779.9 MHz (Used by ED to transmit Join Request)

Infrastruktura založená na technologii LoRa má za úkol sbírat data z nejrozličnějších senzorů, čidel a detektorů, které předávají data prostřednictvím aplikace koncovým uživatelům. LoRa tyto aktivity dokáže zajistit s velmi malou spotřebou elektrické energie, což umožňuje instalovat čidlo takřka kdekoli, aniž bychom řešili otázku napájení. Průměrná životnost na jednu baterii se počítá v řádech několika let. V porovnání s ostatní technologií jakou je Sigfox [11] nebo Zigbee je energeticky mnohem méně náročná a má větší dosah. Veškeré komunikační aktivity jsou v LoRa infrastruktuře koordinovány pomocí takzvané výchozí brány neboli Gateway, které komunikují se síťovým serverem. Tento server pak slouží jako rozhraní pro odesílání a přijímání dat do a z internetu. Díky tomu je možné pomocí našich povelů, nebo automatizovaně reagovat na události a opět je přenášet pomocí infrastruktury LoRa.

2.1 LoRa WAN

Protokol LoRa WAN zajišťuje zabezpečený přenos dat mezi koncovými zařízeními a aplikací běžící na internetu. Každý koncový bod má přidělený jedinečný 64-bitový identifikátor a dva 128-bitové klíče, kterými jsou data pomocí AES zašifrována. Jeden klíč se používá v rámci sítě, to aby mohlo stejné území pokrývat více sítí, druhý k zabezpečení dat mezi sítí a uživatelskou aplikací.

Autentizace a zabezpečení je prováděno pomocí dvou metod. První metodou je Activation By Personalization (ABP), kde jsou klíče nakonfigurovány přímo v zařízení. Jedná se o adresu zařízení *DevAddr*, síťový klíč *NwkSKey* a klíč pro relaci aplikace *AppSKey*. Tímto je zařízení připraveno ke komunikaci se sítí bez dalších postupů. Tato varianta je využita v realizaci této práce. Další metodou zabezpečení sítě je Over the Air Activation (OTAA), znázorněná v diagramu na obrázku č. 2.1. V prvním kroku zařízení přenáší žádost o připojení do aplikačního serveru *Join Request* obsahující globálně jedinečný identifikátor *DevEUI*, identifikátor aplikace *AppEUI*. Zařízení následně obdrží odpověď *Join Accept* od serveru spolu s autentizací koncového bodu a dešifrovanou adresu zařízení. Navíc koncové zařízení obdrží zabezpečující klíče *SecurityKey*, kterými je síťový klíč *NwkSKey* spolu s aplikačním klíčem *AppSKey*.



Obrázek 2.1: Diagram autentizace metody OTAA

2.1.1 Třídy

V současné době jsou definovány celkem tři třídy koncových bodů, které se liší dobou naslouchání a tudíž i vlastní spotřebou. Třída A naslouchá jen chvíli po odvysílání dat, proto má spotřebu nejnižší. Třída B naslouchá po odvysílání dat jako třída A a v plánovaných časech. Poslední třída C naslouchá trvale a kromě doby vysílání má trvalý proudový odběr, tudíž největší.

3 Přehled výrobců LoRa modulů

3.1 Přehled výrobců modulů

3.1.1 Semtech

Společnost Semtech je dodavatelem analogových a takzvaných mixed-signal polovodičů. Z jejich portfolia jsou však nejvíce známé čipy IC pro automobilový a vojenský průmysl, pomocí kterých umožňují bezdrátovou komunikaci. Působí také v oblasti bezdrátových nabíječek LinkCharg a poskytují TVS (Transient Voltage Suppression) diody, které chrání nízkonapěťové obvody před poškozením způsobeným elektrostatickým výbojem (ESD), bleskem a dalším destruktivním napětím. Specifickou oblastí je Power Line Communication (PLC), které umožňují komunikaci přes nízké a vysoké elektrické napětí na vedení. V neposlední řadě se zabývá vývojem a výrobou čipů, podporujících modulaci pro velké vzdálenosti speciálně určených pro IoT sektor.

3.1.2 HOPE Microelectronics

Zaměřují se na vývoj a výrobu různých senzorů a rádiových modulů. Vysílače a přijímače jsou vybaveny podporou pro změnu frekvenční oblasti, díky čemuž je možné velké využití v různých oblastech bezdrátové komunikace. Nabízená řešení v oblasti SmartHome, představují profesionální systém, cenově dostupnější s menšími energetickými nároky. V oblasti IoT nabízí transceivery typu RFM95W pro 868/915Mhz a RFM96W, RFM98W pro 433/470 Mhz, které umožňují bezpečnou a spolehlivou komunikaci. Modul je v SMD provedení s postranními kontakty.

3.1.3 Laird

Tato firma má znalosti v end-to-end systémech a využívá pokročilé nástroje a zkušenosti převážně zaměřené na anténní řešení pro pevné a mobilní klienty, základové stanice, rádia a telematiky. Jejich výrobky se především uplatňují pro veřejnou bezpečnost, dopravní síť a IoT, kde pokrývají trh segmentem produktů, které kombinují různé bezdrátové IoT technologie v jednom modulu s nezbytnou certifikací, která zajišťuje bezpečnou a spolehlivou konektivitu. (Wi-Fi, Bluetooth, BLE, LoRa, Zigbee, point-to-point implementace).

3.1.4 Microchip

Je předním poskytovatelem mikrořadičů a analogových polovodičů. Produkty společnosti jsou pro zákazníky rychle implementovatelné a nabízí tak poměrně levný vývoj. Ve svých produktech má nejmenší zastoupení pro danou technologii LoRa modul, konkrétně RN2483 na frekvenci 433/868 Mhz a RN2093 na 915 Mhz. V modulech je obsažen PIC mikrokontrolér s nízkou spotřebou energie.

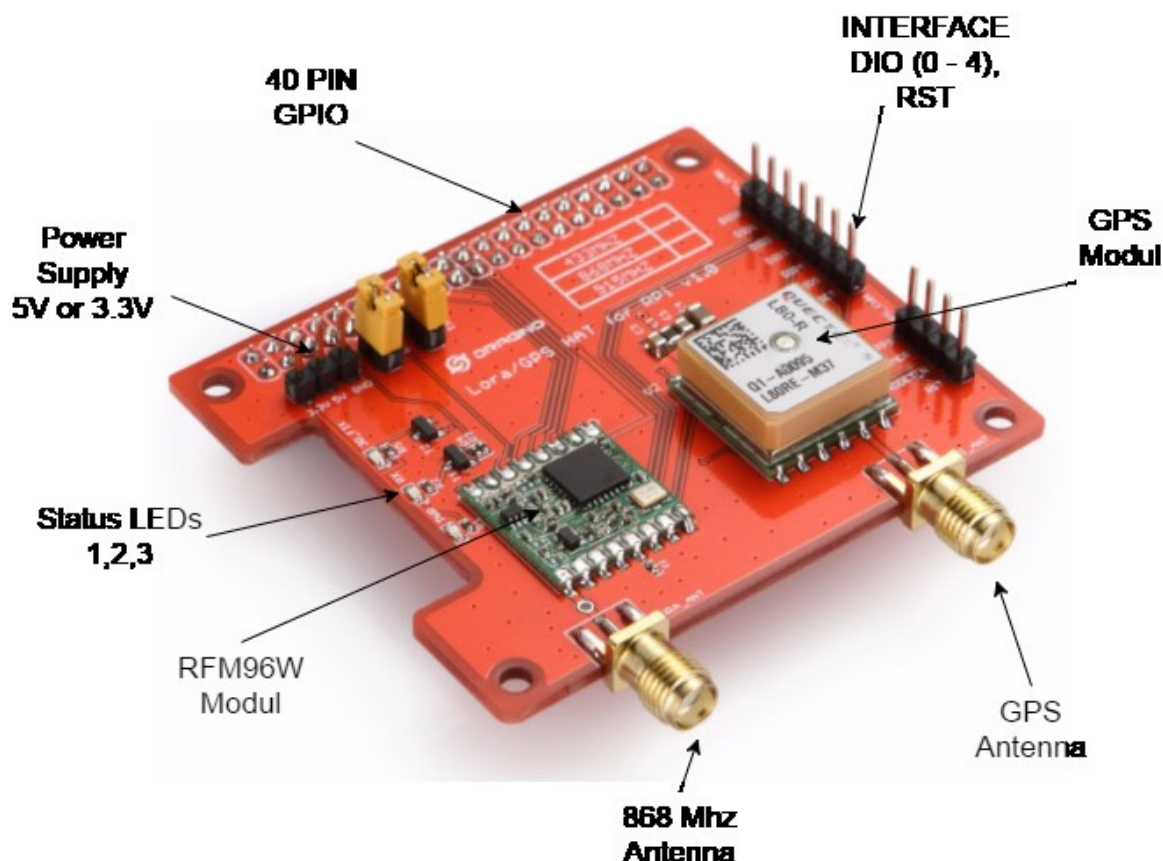
3.1.5 IMTS GmbH

Profesionální centrum pro vývoj vysokofrekvenčních obvodů, bezdrátových modulů a komunikačních systémů. Společnost udržuje partnerství jak na komerčním trhu, tak na poli veřejného výzkumu. Ve svých řadách výrobku má značné zastoupení LoRa WAN, kde poskytují USB a klasické moduly založené na čipech od společnosti Semtech, pomocí kterých sestavují obvody iC880A, které mají uplatnění pro výstavbu profesionálních bran.

3.2 Rozbor modulů

3.2.1 Rádiový modul RFM95W

Rádiové moduly řady RFM95/96/97/98/(W) je možné použít v různých oblastech vzdáleného řízení a automatizace v různých systémech (Automated Meter Reading, Home and Building Automation, Wireless Alarm and Security Systems, Industrial Monitoring and Control, Long range Irrigation Systems) [17]. Moduly jsou v souladu se standardem IEEE 802.15.4 (LR-WPANs) [21] definovaným v roce 2003 **6LoWPAN. Tento transceiver od společnosti HOPE Microelectronics je možné využít jako vysílač (Tx) ale také jako přijímač (Rx). Rádiový modul lze provozovat na frekvenci od 137 Mhz do 1027 Mhz s integrovaným syntetizátorem s rozlišením 61 Hz. Součástí obvodu je modulace typu FSK, GFSK, MSK, GMSK, LoRa TM a OOK. LoRa umožňuje komunikaci na velké vzdálenosti v řádech kilometrů, přičemž umožňuje nastavení šířky rozptýřeného spektra (BW), dále podíl rychlosti přenosu pulzů a rychlosti přenosu znaku (Chip rate / Symbol rate). Čip používá pro své implementace spoustu výrobců, jedním z nich je společnost Dragino, která dodává produkty pro LoRa infrastrukturu založené na těchto čípech. Jedním z produktů je Lora/GPS HAT [16] na obrázku níže č. 3.1, jedná se o nástavbu s vyvedenými piny pro Raspberry Pi [1], navíc je zde integrován GPS modul.



Obrázek 3.1: Dragino shield pro Raspberry Pi

3.2.2 Rádiový modul Semtech SX1276

Moduly Semtech řady SX1276/77/78/79 jsou analogické k transceiverům RFM95/96/97/98(W). Poskytují velký dosah s vysokou odolností proti rušení a zároveň minimalizují spotřebu. Použití modulu používající LoRa modulaci patentovanou firmou Semtech lze dosáhnout citlivosti až -148dBm. V

kombinaci s vysokou citlivostí a integrovaným 20 dBm zesilovačem, je nasazení tohoto modulu vhodné pro rozsáhlé a robustní aplikace.

3.2.3 Radiový modul Laird RM1xx Series

Nabízí řešení pro Enterprise internet věcí (EIoT) od společnosti Laird s inovativní kombinací dvou moderních technologií Bluetooth v4.0 a LoRa WAN s integrací na jednom plošném spoji. Modul je schopen přenášet informace až na vzdálenost 15 kilometrů, překlenuje jak osobní, tak i širokou síť. Vytváří tak scénáře, které dříve nebyly možné, například Smart metering, nebo průmyslová automatizace.

3.2.4 Rádiový modul Microchip RN2483

Je plně certifikovaný 433/868 MHz modul na bázi bezdrátové technologie LoRa. RN2483 využívá unikátní modulaci s rozprostřeným spektrem modulaci v rámci Sub-GHz umožňující maximální dosah, nízký výkon a vysokou kapacitu sítě. Implementovaný LoRa WAN protokol třídy A umožňuje bezproblémové připojení do síťové infrastruktury, ať už veřejné nebo soukromé. Tento modul obdržel jako první certifikaci od LoRa Alliance se specifikací LoRa WAN 1.0.

3.2.5 iU880B - Long Range USB Adapter

Kompaktní a efektivní řešení v podobě USB adaptéru. Umožňuje nastavit výstupní výkon až na +18 dBm (EIRP) v kombinaci s vysokou citlivostí. Podpora ovladačů FTDI-USB pro Linux, Microsoft Windows a WindowsCE. Je možné využít pro velké množství aplikací IoT a M2M. Zařízení se dodává s rozhraním WiMOD LoRa WAN HCI umožňující snadnou konfiguraci. Uvnitř modulu se nachází integrovaná keramická anténa. Maximální zisk této antény je 0,1 dBi.

3.2.6 iC880A - LoRa WAN Koncentrátor

Koncentrátor je schopen přijímat pakety z různých zařízení a odesílat s rozdílným spreading faktorem až na osmi kanálech paralelně. Umožňuje tak příjem několika paketů v jeden okamžik. Koncentrátor lze propojit s jakýmkoliv MCU, propojeným pomocí SPI. Přijaté pakety ukládá do fronty typu FIFO a následně jsou čteny pomocí MCU skrze SPI rozhraní.

Samotný digitální modul SX1301 integrovaný na desce poskytuje 10 programovatelných možností příjmu signálu. Způsoby jsou rozděleny do úrovní od IF0 až IF9, které nám dovolují různé případy užití. Demodulační LoRa kanály IF0 až IF7 mají pevnou šířku pásma kanálu 125 kHz, kterou není možné měnit nebo konfigurovat. Pro každý jednotlivý IF kanál je možné frekvenci libovolně nastavit. Jednotlivé kanály umožňují přijímat několik paketů s použitím různých přenosových rychlostí a faktorem šíření. Kanály jsou dimenzovány pro obsluhu desítek koncových zařízení z hvězdicové topologie.

Kanál IF8 je určen pro LoRu, je propojen spolu s SX1257. Demodulační šířka pásma může být nakonfigurována na 125, 250 nebo 500 kHz. Demodulační cesta je kompatibilní s transmitrem rodiny SX1272 a SX1276 čipů.

4 Gateway

Z definice je gateway neboli výchozí brána, prostředek či rozhraní mezi dvěma různými sítěmi. V tomto případě je toto rozhraní mezi LoRa sítí a sítí internetu, kde propojuje jejich odlišné komunikační protokoly. Brána je sestavena z programu, který má přístup do obou typově různých sítí a to prostřednictvím LoRa jednotky pro komunikaci s LoRa protokolem a internetu pomocí síťové karty. Z předchozích řešerů byl zvolen LoRa Shield od společnosti HOPE s integrovaným čipem RFM96W, jedná se o realizaci jednokanálové brány s výpočetní jednotkou postavenou na Raspberry Pi verze 3 model B. Navrhovaná gateway je mezičlánkem v komunikaci mezi čidlem a vzdáleným serverem popsáným v kapitole č. 5.

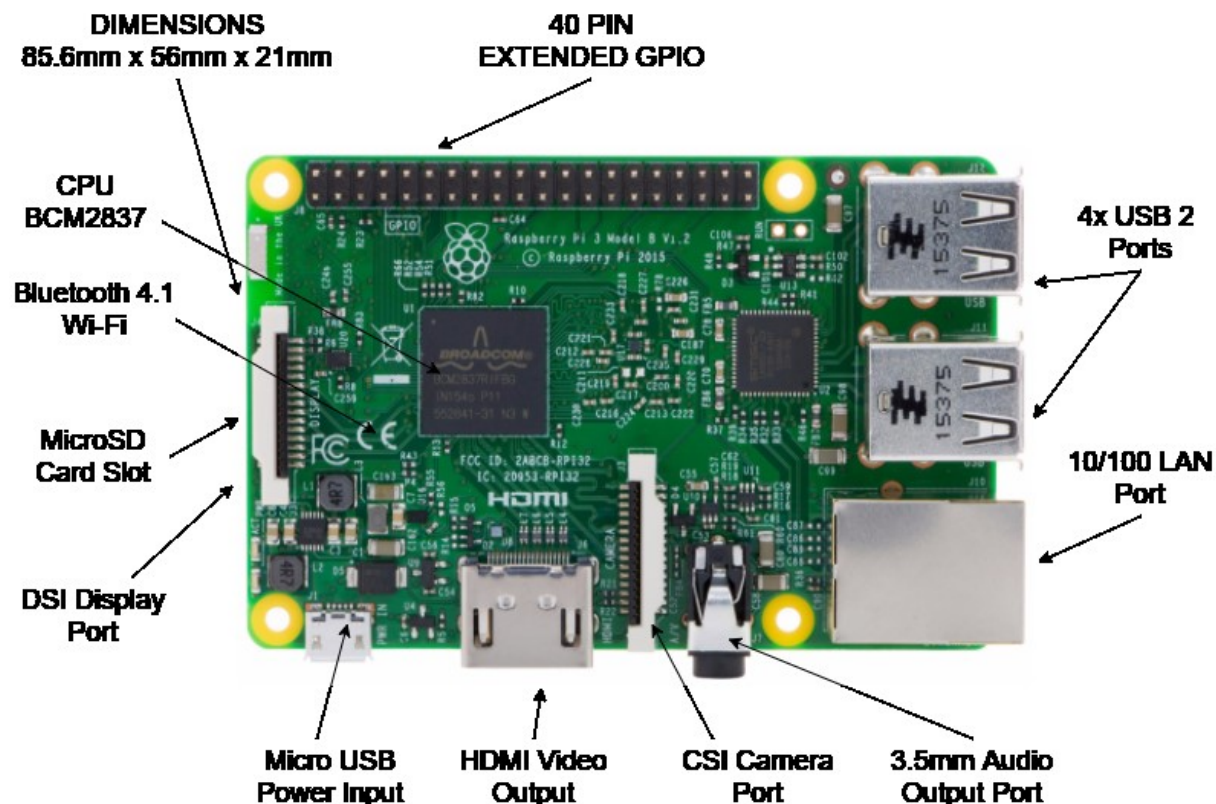
4.1 Raspberry Pi

Tato podkapitola se věnuje jednodeskovému počítačovému modulu Raspberry Pi a jeho využití v této práci. Jedná se o plnohodnotný jednodeskový počítač vyvinutý britskou nadací Raspberry Pi Foundation s cílem podpory výuky informatiky na školách a seznámit studenty s tím, jak mohou počítače řídit různá zařízení. Primárním operačním systémem je Raspbian odvozený z Debianu, který je v této práci pro sestavení brány využit. Jedná se o vysoce optimalizovaný systém pro ARM procesory, využívající RPi. Jako operační systém lze volit z různých systémů. V následující níže uvedené tabulce č. 4.1, je soupis všech podporujících operačních systémů.

Tabulka 4.1: *Operační systémy [32]*

Operační systémy	
BSD	FreeBSD • DragonFly BSD • NetBSD • OpenBSD — Minix 3
Linux (distribuce)	Debian • Fedora • Gentoo • Knoppix • Mandriva • Manjaro • Red Hat • Slackware • Slax • SUSE • Ubuntu
GNU	GNU Hurd • Linux-libre — seL4
OS X	10.6 (Snow Leopard) • 10.7 (Lion) • 10.8 (Mountain Lion) • 10.9 (Mavericks)
DOS	MS-DOS • DR-DOS • Enhanced-DR-DOS • FreeDOS • PTS-DOS
Windows	Windows pro MS-DOS, Windows NT, Windows CE
Mobilní telefony a PDA	Android • Bada • BlackBerry OS • Firefox OS • iOS • Maemo • Palm OS • Symbian OS • Tizen • Ubuntu Touch • webOS • Windows Phone
	QNX • Solaris • BeOS • OpenVMS • Mac OS • NeXTSTEP • Syllable • ReactOS • Haiku • Chrome OS • AmigaOS

V době vzniku této práce již byla uvedena nová verze Raspberry Pi 3 na trh. Tento model je poprvé vybaven 64 bitovým procesorem o taktu 1,2 GHz. Jedná se o čtyř jádrový ARM Cortex-A53 procesor, a podle odhadů výrobců je o 50% rychlejší než v předešlé verzi Raspberry Pi 2. Dále je vybaven 1 GB operační pamětí, s integrovaným Wi-Fi a Bluetooth modulem. Pro RPi vznikli různé rozšiřující desky, jako jsou například rozhraní pro RS232, RF moduly nebo řízení DC motorů. Periferie znázorněné na obrázku č. 4.1 jsou určeny pro spolupráci s výpočetním modulem. Periferie je možné připojit přes HDMI, USB, GPIO, LAN, Audio, Bluetooth, MicroSD, DSI a CSI.



Obrázek 4.1: Raspberry Pi 3, model B

Alternativní moduly, kterými lze nahradit RPi, jsou Banana Pi, AMD Gizmo Board, Intel Galileo, Intel Edison, Tinker Board (Asus).

4.1.1 GPIO Rozhraní

Jedno z rozhraní, kterými RPi disponuje, je GPIO (General-Purpose Input/Output) s čtyřiceti programovatelnými piny, uspořádaných v patici na desce vyobrazené na předešlém obrázku č. 4.1. Jednotlivé piny realizují různé funkce rozhraní, SPI, UART, I2C s napětovými výstupy 3,3V, 5V a GND. Špatná napěťová hladina přivedená na piny může nenávratně poškodit celou desku.

4.2 Komunikace

Úkolem brány je čtení paketů a přeposílání dat na server. Odeslání probíhá pomocí zvoleného komunikačního protokolu MQTT [19] na vzdálený server. Na samotné bráně je k tomuto účelu vytvořen program v programovacím jazyce C++ pro sběr dat ze shieldu. Tyto data jsou čtena pomocí GPIO pinů skrze Raspberry Pi z LoRa shieldu. Pro komunikaci s GPIO byla implementována knihovna *WiringPi* [28]. Jedná se o knihovnu, která je napsaná v jazyce C pro čip Broadcom BCM2835. Tato knihovna je šířena pod GNU LGPLv3 licenci [29].

Následující kód provádí simultánní zápis a čtení do registrů nad vybranou SPI sběrnici (Serial Peripheral Interface) popsanou v kapitole 8.2.

```
data[0] = address
data[1] = data
wiringPiSPISetup(int channel, int speed);
wiringPiSPIDataRW(int channel, unsigned char* data, int length);
```

Pro použití sběrnice je nejprve nutné nainportovat *wiringPiSPI.h* knihovnu. Z této knihovny je využita metoda pro čtení a zápis *wiringPiSPIDataRW*, jejíž vstupními parametry jsou kanál, data a délka dat. Samotný kanál je inicializován metodou *wiringPiSPISetup*, RPi disponuje dvěma kanály (0 a 1), parametrem speed je pak celočíselná hodnota rychlosti hodinového impulsu tento parametr lze nastavit v rozsahu od 500 000 až do 32 000 000 Hz. Návratová hodnota je deskriptor zařízení, v případě chyby je vráceno -1. První pole dat musí obsahovat adresu, do kterého registru přistupujeme. Za ním následuje pole s daty, která chceme odeslat nebo samotná přijatá data. Například pro přístup k datovému bufferu na LoRa modemu je definován registr, který určuje bod v paměti. Ve výchozím nastavení je určena polovina dostupné paměti pro RX inicializované na adrese 0x00 a druhá polovina je určená pro TX inicializované adresou 0x80. Princip operací ostatních registrů je podrobně popsán v příloženém datasheetu pro čip RFM96W od společnosti Hope. Následně jsou vyčtená data parsována a obalena do json obálky.

Takto zpracovaná data jsou předána do metody *sendDataToBroker* pro odesílání zprávy na MQTT server. Pro tento účel je zde implementovaná knihovna eclipse paho [19]. Jedná se o open-source řešení, pro klientskou implementaci MQTT a MQTT-SN zaměřeného na nově vznikající aplikace pro internet věcí. MQTT od roku 2016 spadá pod ISO standard ISO/IEC 20922 [20]. Detailní popis MQTT je uveden v kapitole č. 5 serverové části. V následující metodě je prvním vstupujícím parametrem msg neboli zpráva v json obálce spolu s payloadem, ke kterému je přidělen topic (odběr).

```
sendDataToBroker(char *msg, char *topic)
```

Tento topic je generován ve tvaru */APP/* id aplikace */GW/* mac adresa brány a */DV/* id zařízení v případě že se nejedná o zašifrovanou LoRa WAN zprávu metodou ABP. V opačném případě se ID zařízení neurčí a zpráva se dešifruje na serveru. Pro jednoduché uživatelské nastavení bylo vytvořeno konfigurační prostředí. Jedná se o konfigurační soubor *lora-gw.conf*, ve kterém se nacházejí jednotlivé parametry sítě a přístupové údaje ke vzdálenému serveru.

```
# Set center frequency in 868000000 Hz (868 Mhz)
set_center_frequency 868100000

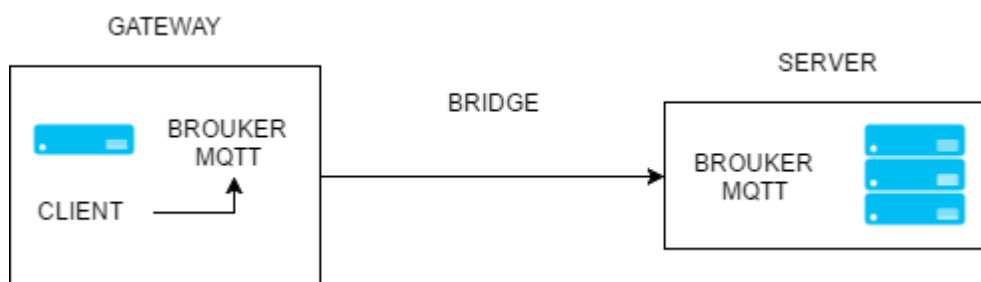
# Set location GPS module
set_location_lat 49.83156
set_location_long 14.16123
set_location_alt 0
```

Gateway

```
# Set contact information
email gre0071@vsb.cz
description LoRaWAN GW001 popis technologie

# Set parameter for MQTT
ms enable
ms_address tcp://localhost:1883
ms_client_id GW002-A
ms_password gateway
ms_username gatewaytest
ms_qos 1
ms_timeout 10000L
app_euis 816C4E6E36849145
```

Komunikaci se serverem lze nastavit dvěma způsoby, a to díky implementovanému brokeru Mosquitto [19]. První variantou je propojení dvou brouků (MQTT Serverů), kdy vytvořením tohoto spojení zhotovíme mezi body takzvaný bridge. Klient nekomunikuje přímo se serverem ale se svým lokálním broukem (agentem). Popis je znázorněn pomocí diagramu na obrázku č 4.2. Výhodou tohoto řešení je záloha dat, v případě ztráty spojení se serverem, kdy po opětovném obnovení komunikace se předešlá data znovu odešlou do brouku na serveru. Na takto zvolenou variantu je potřebné v nastavení brány změnit adresu MQTT klienta na localhost a restartovat brouka.



Obrázek 4.2: *MQTT komunikace využívající bridge*

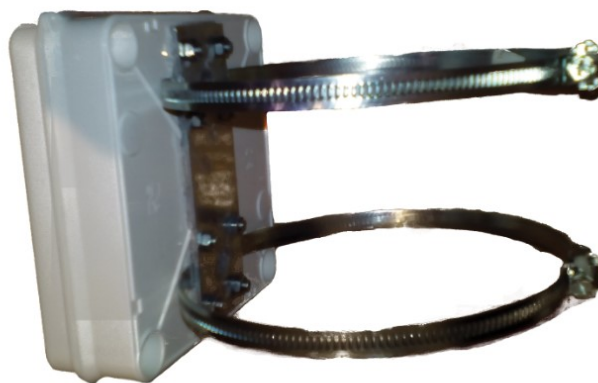


Obrázek 4.3: *MQTT komunikace Server a Klient*

Další variantou je přímá komunikace se serverem a klientem vyobrazená na obrázku č. 4.3, kde je nastavena adresa MQTT přímo na server s TCP portem 1883. Brána byla zprovozněna a otestována pod systémem Raspbian, na kterém jsou implementovány veškeré potřebné náležitosti pro chod brány. Z takto rozšířeného systému o LoRa bránu byl vytvořen image a pojmenován jako Lorbrian, který tak lze jednoduše instalovat a šířit na další RPi zařízení. Toto rozšíření bylo také implementováno jako balíček pro OpenWrt [26].

4.3 Konstrukce brány

Jedním z požadavků, který vyplývá z vystavení komponentů pro bránu na venkovním prostředí, je odolnost brány vůči vnějším vlivům. Brána je zhotovena v montážní krabici s pokrytím IP56. Rozměry této krabice jsou 190 mm x 140 mm x 70 mm, což plně postačuje pro jednotlivé komponenty použité pro bránu, kterými jsou Raspberry Pi 3 model B, LoRa Shield od společnosti Hope, Bleskojistka, SAW filtr a napěťový regulátor. Uchycení brány je provedeno dvěma kovovými stahovacími páskami, tyto pásky jsou uchyceny do krabice v sendvičové konstrukci, pomocí dvou předvrtaných kovových plátů o tloušťce 2,5mm. Tyto pláty jsou spojeny pomocí čtyř maticových šroubů. Svorník s plochou kruhovou hlavou o průměru 8,79 mm, délce 24 mm, je opatřen metrickým závitem M5. Mezi tyto kovové pláty byl aplikován univerzální průhledný silikon pro utěsnění proti vodě. Výsledná konstrukce je vyobrazena níže na obrázku č. 4.4.



Obrázek 4.4: *Montážní krabice pro bránu.*

Pro napájení gateway, je důležité stejnosměrné napětí o velikosti 5 V. Zdroj tohoto napětí, je řešen pomocí PoE (Power over Ethernet), které má výstupní napájení 48V. Toto napětí bylo na konci uzlu regulováno na 5V a to pomocí regulátoru AP K85, s výstupním napětím 5 V a maximálním protékajícím proudem 1.5A



Obrázek 4.5: *Regulátor napětí, bleskojistka a SAW filtr.*

Ochrana zařízení proti výboji blesku do antény, byla provedena pomocí bleskojistky vložené, jakož to mezičlánek přívodu koaxiálního kabelu k modulu. O příjem signálu se stará prutová anténa Huawei E620/1823/3131/3372, 5 dB, která je navržena pro příjem na GSM/3G pásmu, z tohoto důvodu je zde použit předřadný SAW filtr, který odfiltruje nežádoucí pásma. Pro filtraci je použit SMD SAW filtr EPCOS s 6 Piny se střední frekvencí 868MHz. Jako paměťové médium v zařízení je použita microSD karta o velikosti 8GB, kde je nahrán image systému Raspbian,

5 Server

Z definice je server označením pro počítač, který poskytuje služby a program realizující dané služby. Server tyto služby poskytuje následně klientům. V tomto případě se jedná o dedikovaný server běžícím na virtuálním stroji, pod linuxovou distribucí Debian 8 Jessie. Server je připojen do školní sítě VŠB-TUO a má zvenčí přístupné porty 80 pro HTTP, 443 pro HTTPS a dva vyhrazené porty pro MQTT 1883 a 1884 Websocket. Pro vizuální zobrazení dat a klientskou správu údajů je na serveru implementována webová aplikace popsaná v kapitole číslo 6. Součástí práce je také webová prezentace LoRa technologie umístěné na serveru.

5.1 Komunikace

Jedna z implementací na serveru je Mosquitto MQTT broker, který zprostředkovává komunikaci mezi serverem a bránou. Samotný popis MQTT je popsán v kapitole 5.1.2. Před příchodem zprávy je doručující klient ověřen pomocí ACL popsaného níže v kapitole 5.1.1. Dále je zde vytvořen klient, takzvaný koncentrátor v jazyce Python, pomocí kterého je přihlášen na lokálním broukru. Klient je implementací Paho MQTT knihovny. Na koncentrátoru tento vyhrazený klient čeká na zprávu s jakýmkoliv adresátem. Po přijetí zprávy je identifikován odesílatel a následně jsou v databázi dohledány uživateli přidělené klíče pro dekódování payloadu obsaženého v json obálce. Takto přijatá zpráva je následně uložena spolu s identifikujícím záznamem do databáze. Pro zvýšení bezpečnosti je možné tyto data ukládat s payloadem v zašifrované formě a samotné dešifrování payloadu provést až na straně aplikace.

5.1.1 ACL

ACL (Access Control List), je jedním z prvků základního zabezpečení. Jedná se o seznam oprávnění přiděleného uživateli pro práci s určitými mechanizmy. Kontrola spočívá v prohledání seznamu oprávnění před každou prováděnou operací. Po nalezení odpovídajícího záznamu pro daného uživatele, rozhodne podle tohoto záznamu, zda operaci může nebo nemůže provést. Seznamem je myšlena tabulková struktura obsahující specifikující práva uživatelů nebo skupin. V tomto případě je pro ACL využit open source plugin mosquitto-auth-plug dostupný z GitHub. Plugin je doplňkem pro mosquitto a zajišťuje tak vstupní kontrolu přijatých zpráv, a to formou autentizace a autorizace. Seznam oprávnění je vytvořen v MySQL [7] databázi serveru společně s přístupovými údaji brán a aplikací. Údaje reprezentují uživatelské jméno neboli EUI aplikace či brány v 64bitové podobě a zahešované heslo pomocí funkce SHA-2 s 256 bity. Takzvaný back-end pro autentizaci lze nastavit na MySQL, PostgreSQL, CDB, SQLite3 database, Redis key/value store, TLS PSK, LDAP, HTTP (custom HTTP API), JWT a MongoDB. Parametry pro oprávnění jsou ID, TOPIC a RW. RW znamená, zda uživatel má právo v rámci topiku zapisovat (1), číst (0) nebo zapisovat i číst (2). U uživatele lze pak nastavit parametr, zda je takzvaný super user, umožňující zápis a čtení nad všemi topikami. Výsledná tabulka je znázorněna v relačním modelu databáze.

5.1.2 MQTT

Základním prvkem komunikace mezi serverem a gateway je MQTT protokol, skrze kterého jsou data obalena v json obálce. V prvním kroku navazuje klient spojení s broukrem pomocí TCP. Nejčastěji se používá pro komunikaci port 1883, pro spojení s TLS pak 8883. Výhodou je možnost využití

WebSockets, který je možné jednoduše rozkomunikovat přes JavaScript do webových stránek. Samotnou komunikaci s nastavením portů lze nakonfigurovat podle vlastních potřeb.

V druhém kroku po navázání spojení odešle zařízení zprávu CONNECT. V případě že je nastaven *clean session*, jsou vyčištěné veškeré odběry v opačném případě, se provede připojení bez *clean session*, což bude znamenat, že veškerá dříve přihlášená témata zůstanou. Přihlášení na server probíhá pomocí přihlašovacího jména a hesla, které je vygenerováno skrze webového prostředí pro gateway. Server odpovídá zprávou CONNACK, o potvrzení připojení brány. Zprávou SUBSCRIBE společně s názvem záznamu se přihlásíme k tématu, které chceme odebírat. Server potvrdí odběr zprávou SUBACK. Následně může server a gateway odesílat zprávy pomocí PUBLISH. Odhlášení od odběru se provede zprávou UNSUBSCRIBE a následně server potvrdí zrušení odběru tématu zprávou UNSUBACK. Témata jsou kontrolována pomocí ACL, které určí jaký topic může daný uživatel používat či nikoli. Tato ochrana slouží k tomu, aby nemohl druhý uživatel přijímat či odesílat zprávy na cizí zařízení. Odhlášení se provede zprávou DISCONNECT, který zajistí ukončení spojení.

MQTT dokáže detekovat, zda je zařízení online nebo off-line. Pokud je zařízení dlouhodobě neaktivní a nevykazuje žádnou aktivitu, považuje se za odpojené. Tento stav se prověřuje odesláním zprávy ze zařízení PINGREQ kdy server-broker odešle potvrzující zprávu PINGACK. V případě ukončení spojení odešle takzvanou závěť *last will*, která je definována při vytvořeném spojení se serverem CONNECT např. DEVICE OFFLINE.

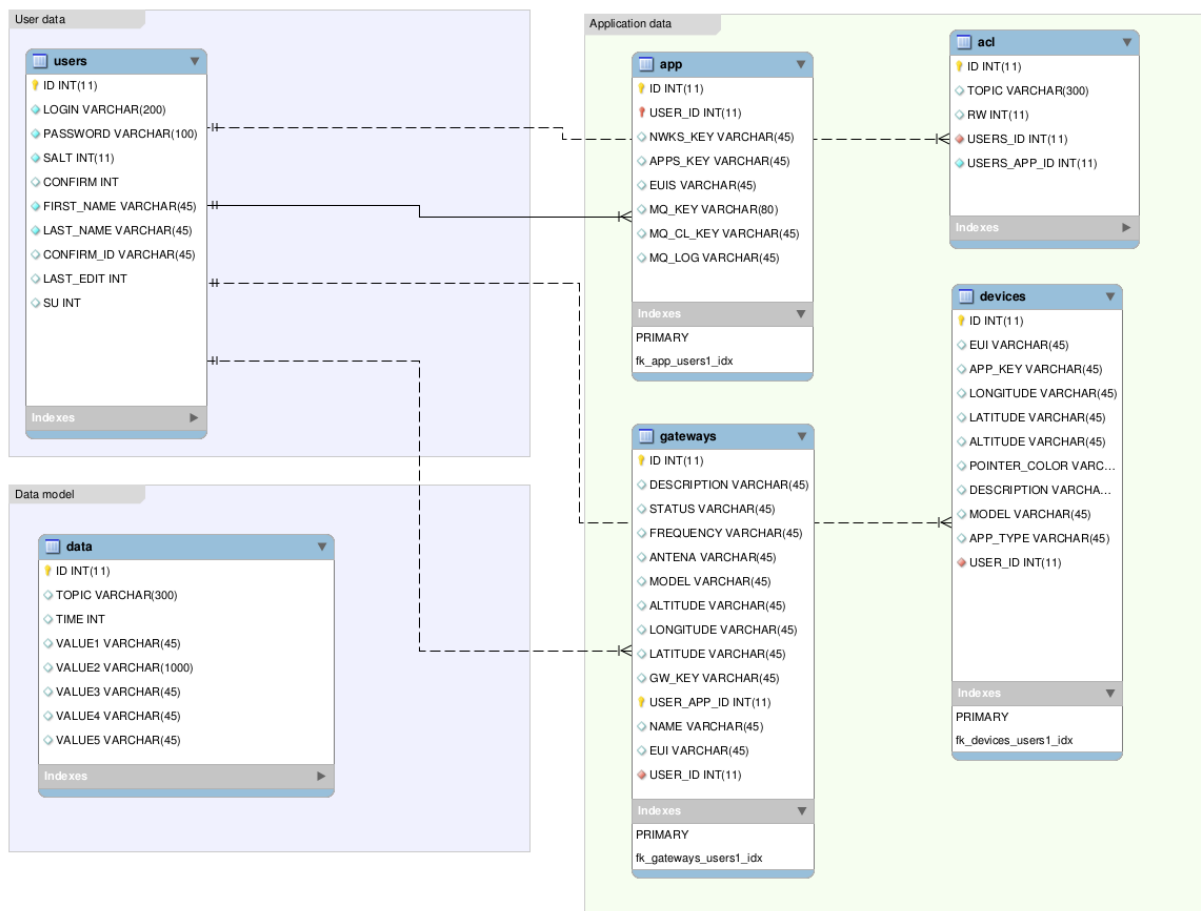
5.1.2.1 *Quality of Service*

Existují tři úrovně 0, 1 a 2, které jsou rozděleny podle toho, jak jsou potvrzené a podle zajištění předání zprávy. Na nulté úrovni odesílá publisher jen zprávu PUBLISH brokeru. Pro první úroveň posílá publisher brokeru zprávu PUBLISH a následně čeká na odpověď. Broker přijme zprávu a následně ji přepoše ostatním uživatelům, kteří odebírají dané téma. Odesílání zprávy a nastavení QoS probíhá podle toho s jakou úrovní, byla QoS zpráva přijata. Broker odesílá zprávu s QoS na takové úrovni na které zpráva přišla, maximálně s nižší úrovní QoS než disponuje samotný broker, pokud příjemce danou úroveň nepodporuje. Na druhé úrovni posílá publisher brokeru zprávu PUBLISH, stejně jako v případě první úrovně QoS 1, ji přijme, přepoše odběratelům a následně odešle zpět potvrzení o publikování PUBREC. Publisher následně odpoví PUBREL, čímž broker zprávu smaže a potvrdí PUBCOMP.

5.2 Databáze MySQL

Relační databázový server MySQL, vznikl jako firemní projekt. Během pár let se stal tak populární, že na něj vytvořili vlastní společnost zaměřenou na MySQL a jeho služby. Řešení nabízí stejné vlastnosti nalezené v konkurenčních produktech. Jedná se o vysoce optimalizovaný produkt dostupný pro různé platformy Compaq Tru64, DEC OSF, FreeBSD, IBM AIX, HP-UX, Linux, Mac OS X, Novell NetWare, OpenBSD, QNX, SCO, SGI IRIX, Solaris (verze 8, 9 a 10) a Microsoft Windows. Komunikace s databází probíhá pomocí jazyka SQL. Podpora obrovské komunity a široká škála dostupných API pro všechny nejpobulárnější programovací jazyky C, C++, Java, Perl, PHP, Ruby a Tcl. Mezi známé uživatele patří Yahoo!, NASA, Google a Cisco Systems. MySQL se nabízí pro mnoho použití zcela zdarma. Také nabízí mnoho typů mechanismů pro správu dat, jako úložné systémy. Tyto systémy je potřeba správně zvolit s důrazem na použití vhodného algoritmu pro konkrétní úkol. S podporou několika enginů, a to MyISAM (výchozím pro všechny operační systémy krom Microsoftu),

MEMORY (také známe jako HEAP), InnoDB (výchozím pro operační systém Windows) a MERGE. V nedávné době vyšla nová alfa verze engine Falcon, jedná se o vysoce účinný engine pro rozsáhlé nasazení na více vláknových systémech.



Obrázek 5.1: Relaçní model databáze

Pro vytvoření databáze byl využit nástroj MySQL Workbench. Základním krokem bylo sestavení relačního modelu databáze uvedeného na obrázku číslo 5.1. Databáze je rozdělena do šesti tabulek a to **users**, **app**, **acl**, **gateways**, **devices** a **data**. Pro práci s databází jsou vytvořeny objekty pomocí vlastního generátoru napsaného v jazyce Java. Tento generátor pracuje s SQL kódem pro vytvoření databáze vyexportovaného z Workbench. Generátor následně vytvoří pro každou tabulku základní objekt `_Object` spolu s dotazovacím objektem `_Query`. Výstupní objekty lze vygenerovat do jazyka PHP nebo Java. Dotazy obsahují základní práci s databází jako je `update`, `delete`, `create`. Další požadované funkce je nutné doprogramovat. V základním objektu je pak obsažena kontrola `protectFields` proti injectu.

5.3 LAMP

Jedná se o souhrn open-source technologií používané jako platforma pro implementaci dynamických webových stránek. Použitými technologiemi jsou Linux, Apache, MariaDB nebo MySQL, PHP, Perl nebo Python. V této práci byla na server nasazena tato varianta, skládající se z Apache 2, MySQL a PHP 5. LAMP [31] je zde z důvodu webové aplikace, díky čemuž je získán prostředek pro práci s databází.

6 Webová aplikace

Výsledkem této webové aplikace je vizualizace uživatelských dat ze senzorů, správa zařízení a výchozích brán. Na řešení byl kladen důraz, aby výsledná aplikace byla pro uživatele přehledná a jednoduchá z hlediska ovladatelnosti. Aplikace se nachází na straně serveru a je napsaná v jazyce PHP, s využitím níže uvedených technologií v kapitole 6.1. Pro webovou aplikaci byla vybrána administrační šablona SB Admin 2 [34] vytvořena na populárním frameworku Bootstrap využívající HTML, CSS a JS. Framework je určen pro vývoj responzivních stránek, které lze prohlížet na mobilních a desktopových zařízeních s podporou webového prohlížeče. Použitá šablona je nabízena zcela zdarma a bez poplatků, umožňuje tak zrychlit vývoj pro použití ve vlastních aplikacích. Samotná kostra šablony byla graficky upravena a nasazena ve spojení se serverovou částí. Hlavním prvkem webové aplikace pro LoRu je levé postranní menu, pomocí kterého lze přecházet do jednotlivých úrovní stránek, těmito stránky jsou Payload, Gateways, Devices, Application a Maps. Pro vizualizaci grafů je zde použita JS knihovna *Chart.js*, načítání dat do grafu je provedeno pomocí Ajax [5] dotazu na server který vrátí data a přes skript je následně přidá do samotného grafu. V pravém horním menu je uvedena nabídka se jménem uživatele účtu pro nastavení a odhlášení. Proti podvržení odhlášení z účtu je zde vytvořen speciální generující se token pojmenovaný *usskey*, který je spolu s požadavkem na odhlášení odeslán na server. Tím je zajištěno korektní odhlášení od účtu. Pro přihlášení uživatele do aplikace slouží přihlašovací formulář, do kterého je nutné vložit svůj login a heslo. Login představuje emailovou adresu zadanou při registraci. K Samotnému heslu je pak zahájena kontrola na serverové straně, kde je provedeno porovnání hesla z databáze a proveden algoritmus pro kontrolu hesla s vygenerovaným *sold*. Heslo je uloženo pomocí hashovací funkce SHA-2, tento algoritmus je vylepšením SHA-1 a je založen na podobném principu jako je bloková šifra například z AES. Poskytuje výstupní délky 224, 256, 384 nebo 512 bitů. Z důvodu bezpečnosti, kde SHA-1 byla kompromitována kryptografickými útoky, byla zvolena SHA-2, na kterou doposud nebyly zveřejněny žádné úspěšné útoky.

6.1 Technologie pro realizaci webové aplikace

Správná volba technologií pro sestavení aplikace je velmi důležitá pro sloučení ostatních technologií implementovaných na serveru. Takovým příkladem může být technologie PHP, díky které dokáže aplikace provádět dotazování skrze skriptů na databázový server MySQL. Níže uvedené technologie byly využity pro vývoj webové aplikace.

6.1.1 HTML

Hlavním jazykem pro tvorbu stránek je HTML (Hyper Text Markup Language) [3]. Tento značkovací neboli tagovací jazyk, využívá k propojení mezi webovými stránkami hypertextových odkazů. Díky tomuto jazyku je možné publikovat své stránky do celosvětové sítě WWW (World Wide Web). Předchůdcem jazyka je univerzitní značkovací jazyk SGML (Standard Generalized Markup Language). Samotný jazyk je ovlivněn neustálým rozvojem prohlížečů a jejich rozšíření. Aktuální verzí je HTML 5, která ukončuje závislost jazyka na SGML a opravuje chyby z předešlé verze. Dochází k

vyřazení nepoužívaných prvků a přidává nové sémantické. Oproti předchozí verzi tak přináší například tato zlepšení:

- Zjednodušení kódování stránky a zápisů
- Nová pole formulářů a jejich validace, placeholder
- Výběr z více možností (datalist)
- Grafické prvky (svg, canvas)
- Video, audio
- Strukturální rozdělení stránky
- Javascriptem dostupná API (drag & drop, geolokace)

Koncepce jazyka je charakterizována jako množina tagů a jejich atributů. Mezi hranaté závorky je vložena sémantika, součástí mohou být vnořené prvky. Tyto značky rozdělujeme na párové a nepárové. Správný HTML dokument má předepsanou strukturu, která je tvořena z deklarování typu dokumentu DOCTYPE, reprezentaci celého dokumentu kořenovým elementem html, hlavičkou head s metadaty a tělo dokumentu body.

6.1.2 CSS

Kaskádové styly neboli CSS (Cascading Style Sheets) [3] jsou styly pro definování způsobu vyobrazení elementů na webových stránkách. Tyto styly mohou například definovat zarovnání, barvu, velikost, pozici nebo pozadí. Syntaxe se skládá z kolekce pravidel. Každé pravidlo se pak skládá ze selektoru a deklarace. CSS má čtyři verze a to CSS 1, CSS 2.1, CSS 3 a CSS 4. Některé prohlížeče podporují už nové funkce pro CSS, které ještě nebyly standardizovány. Níže je uvedeno pravidlo pro libovolný prvek odpovídající <h1> který obarví text do modré barvy a přidá okraj na levé straně.

```
h2 {  
    color: blue;  
    margin-left: 25px;  
}
```

Za každým pravidlem je nutné uvádět středník pro jejich oddělení. Pravidla stylů je možné umístit přímo do dokumentů, nebo do samotného souboru. Avšak tento samotný soubor musí být do stránky načten a to pomocí linku. V praxi je samostatný soubor využíván k rychlejšímu načtení stránky, kde se opakující elementy nemusí znova načítat a použije se stejný zdroj.

6.1.3 XML

XML (Extensible Markup Language) [3] původní zkratka byla pod označením EML. Umožňuje vytvářet značkovací jazyk pro různá data. Jeho časté využití najdeme v serializaci dat. Dalším možným uplatněním je prezentovat low-level data, například pro konfiguraci. XML neobsahuje vlastní předefinované značky. Tyto značky je možné nadefinovat v souboru DTD (Document Type Definition), což umožňuje typovou kontrolu definic nad dokumentem. Při vývoji je používán takzvaný parser, který

za nás řeší kontrolu a detekuje tak chyby v kódu. Hypertextové odkazy lze vytvořit v rámci dokumentu i navzájem mezi dokumenty. Specifikace odkazů je popsána v následujících standardech XPath (XML Path Language), XPointer (XML Pointer Language) a XLink (XML Linking Language). Syntaxe kódu je závislá na struktuře a integritě.

6.1.4 PHP

Vývoj programovacího jazyk PHP [15] byl zaměřen na obohacení statických HTML stránek o dynamické skriptování. Před příchodem PHP se používaly pro operace programy, které se volaly pomocí CGI (Common Gateway Interface). K tomu, aby se mohl program spustit, musel server vytvořit nový proces. Tím docházelo k velké zátěži operačního systému. PHP je rozděleno do modulů, které jsou načteny jako součást webového serveru. Tím je dosaženo v případě velkého vytížení stránek enormního zrychlení. Dalšími jazyky, které lze implementovat do webového serveru jsou Perl, Python nebo Ruby. Soubor pro zpracování v PHP má koncovku *.php*, tento soubor se následně ověří za pomoci interpretera a provede základní instrukce. Po zpracování výsledků serverem se obsah zobrazí klientovy. PHP podporuje velké množství knihoven například pro zpracování textu, grafů, přístup k databázím (MySQL, PostgreSQL, ODBC, Oracle, MSSQL, Sqlite).

6.1.5 JavaScript

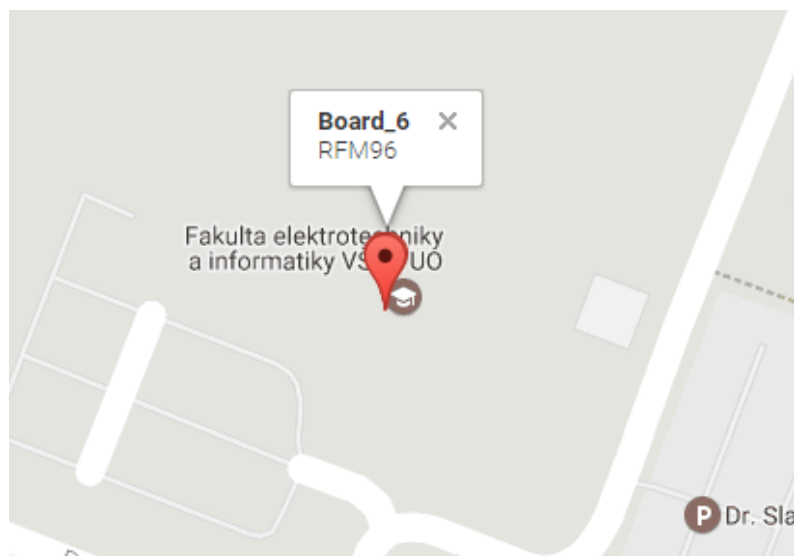
Jedná se o multiplatformní, objektově orientovaný skriptovací jazyk. V současné době se používá jako interpretovaný programovací jazyk webových stránek. JavaScript [4] umožňuje neustálou komunikaci se stránkou, aniž by se musela stránka znovu načíst. Je také důležitým prvkem pro grafickou adaptaci stránky. I když je to názvoslovně zavádějící tak JavaScript a Java nemají téměř nic společného, název těží z popularity Javy. V praxi platí dva termíny ECMAScript a JavaScript, jsou to dvě jména pro stejný jazyk. Webové prohlížeče nejsou jedinou platformou, kde se JavaScript používá. Další platformou může být například některé databáze MongoDB a CouchDB, používají JavaScript pro jejich skriptování a dotazování. Také několik platform pro desktopové a serverové řešení, zajímavým projektem je programování serveru pomocí *Node.js*. V této práci je navíc využita jQuery [5][6] knihovna. Rychlá a malá knihovna pro JavaScript. Dokáže manipulovat a ovládat prvky v HTML dokumentu. Zpracování událostí, animace a zjednodušení Ajaxu.

6.2 Mapy

Pro mapové podklady je zvoleno API od společnosti Google. Google Maps API je k dispozici pro Android, iOS, webový prohlížeč a HTTP webové služby. Obsahem je celá řada tutoriálu a příkladů pro práci s API. Účelem mapy je určit polohu všech uživatelů dostupných zařízení a zobrazení těchto bodů v mapových podkladech. Tato data jsou dotažena z databáze, ze které se získávají údaje o poloze, názvu a popisu čidla. Bylo využito řešení API pro webové stránky založené na JavaScriptu. Prvním krokem pro vývoj je získat takzvaný klíč pro přístup k API. V níže uvedeném kódu je pak tento klíč nahrazen za YOUR_API_KEY.

```
<script async defer
  src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap">
</script>
```

Získat klíč je možné díky založení developer Google účtu, pomocí kterého lze přidávat různé aplikace využívající různé Google API. V druhém kroku byl v jazyce PHP vytvořen generátor XML dokumentu, který stahoval požadovaná data z databáze a následně z nich vygeneroval dokument formátu XML. Pro vložení XML do mapy byl použit skript s Ajaxem spolu s parametry pro generování. Dále tyto data byla rozparsována a přidána do mapy jako jednotlivé body vyobrazené pomocí markeru. V následujícím obrázku č. 6.1, je možné pozorovat jeden z vytvořených bodů.



Obrázek 6.1: Zobrazení senzoru v mapě.

6.3 Správa brán a zařízení

Záložka Gateways je určena k zobrazení veškerých brán, které uživatel spravuje. Vizualizace je provedena pomocí tabulky na obrázku č. 6.2, ve které nalezneme základní informace jako je Název, EUI, Frequence, Model, Position a Status. Dále je na stránce uvedeno tlačítko Add Gateway. Toto tlačítko odkazuje na formulář, který slouží k přidání další brány do aplikace. V tomto formuláři jsou vyžadovány údaje pro bránu a to Name, Description, Range, Latitude, Longitude, Altitude, Frequency, Model, Ready a Antena. Po registraci jsou vygenerovány údaje pro přihlášení brány na MQTT server.

Gateways section

+ Add Gateway					
Name	EUI	Frequency	Model	Position	Status
GW001	1	868 Mhz (EU)	RFM96	49.831470N, 18.158713E	✓
TGW01	3	868 Mhz (EU)	RFM96	49.844950N, 18.153333E	✓

Obrázek 6.2: Správa brány







Další záložkou je Devices, která se rozděluje do dvou bloků. Prvním blokem je tabulka, ve které se nachází veškerá uživatelská zařízení se základními údaji Description, EUI, App type, Model a

Position. Jednotlivé řádky tabulky jsou interaktivní. Po kliknutí na řádek v tabulce se z databáze načnou payloady a to posledních 23 údajů. Zároveň se změní odběr tématu na MQTT Klientovy.

Devices section

+ Add Device

APP [816C4E6E36849145]

Description	EUI	App type	Model	Position	
Board_1	02D1E520	TEMPERATURE	SX1781	49.8311236N, 18.1606757E	 
Board_2	02D1E521	TEMPERATURE	SX1781	49.8311236N, 18.1606767E	 
DP	P1E6EF00	SHOCK	SX1781	49.8311216N, 49.6679664E	 

Select device EUI- P1E6EF00

Data

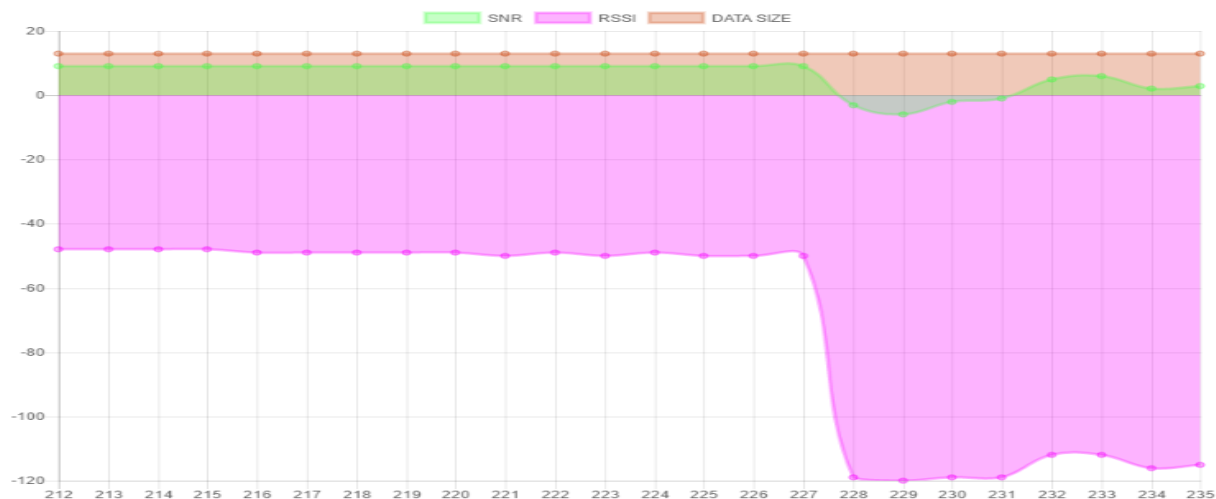
Graph

Payload - application data

Time	Payload	Length	RSSI	SNR
Tue, 11 Apr 2017 10:49:34 GMT	Shock	13	-119	-1
Tue, 11 Apr 2017 10:49:33 GMT	Shock	13	-119	-2
Tue, 11 Apr 2017 10:49:32 GMT	Shock	13	-120	-6
Tue, 11 Apr 2017 10:49:32 GMT	Shock	13	-119	-3

Obrázek 6.3: Správa zařízení

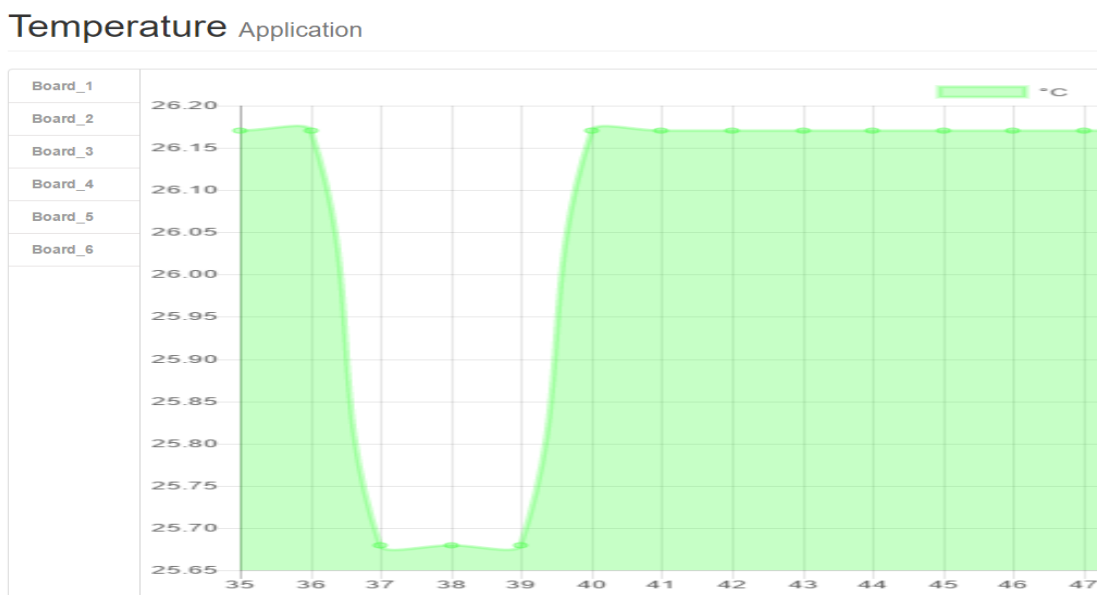
Tím se docílí zobrazení aktuálních dat. Tyto data jsou zobrazena v druhém bloku, kde je tabulka s jednotlivými řádky a informacemi o Payload, Time, Length, RSSI a SNR. Data jsou seřazena sestupně podle času. Jednotlivé tabulky s údaji jsou uvedeny na obrázku č. 6.3. Z informativních údajů o přijatém signálu na gateway je zde vytvořena záložka ve spodním bloku Select device s názvem Graph. V tomto grafu jsou zobrazeny informace o SNR, RSSI a DATA SIZE z posledních 23 přijatých zpráv vyobrazeném na obrázku č. 6.4.



Obrázek 6.4: Graf informující o signálu v době přijetí zprávy

6.4 Aplikace senzorů

Nápadem pro obohacení vizualizace a zpracování dat je vytvořit modulární aplikace, které lze k jednotlivým typům zařízení přidělovat. Jedna z aplikací je určená pro senzory, které jsou schopny měřit teplotu. Po spuštění si aplikace požádá o seznam dostupných zařízení, která podporují aplikaci TEMPERATURE. Požadavek je odeslán do databáze s dotazem a následně server vrátí všechny vyhovující záznamy. Tento seznam je poté zobrazen v levém menu aplikace vyobrazeném na obrázku č. 6.5. Při výběru zařízení ze seznamu se nám pomocí identifikátoru načnou data z databáze pro vybraný senzor, v tomto případě se jedná o teplotu ve stupních celsia. Podobně jako v kapitole 6.3 se zároveň změní odběr tématu na vybrané zařízení pro MQTT Klienta. Díky tomu je možné vidět reálný průběh teploty na čidlu bez nutnosti aktualizace dat z databáze.



Obrázek 6.5: Zobrazení dat v aplikaci Temperature

V grafu na obrázku je na jedné ose vyobrazena teplota ve stupních celsia a na druhé, pořadové číslo záznamu teploty. Na grafu je vždy zobrazeno posledních 23 záznamů.

7 Rozdělení senzorů

Senzor je část elektronického obvodu, jehož chování je závislé na sledované fyzikální veličině. Senzor obsahuje snímací čidlo pro první zpracování signálu, tento převod se většinou rozděluje na více úrovní zpracování. Sensory rozdělujeme podle chování výstupu na senzoru, buďto jako zátěž s definovanými parametry nebo jako zdroj signálů.

7.1 Aktivní senzory

Převodem měřené veličiny z mechanické, tepelné, chemické nebo světelné energie na elektrickou se senzor chová jako zdroj elektrické energie. Aktivní metodou jsou zdroje napětí založené na daném transformačním jevu.

7.2 Pasivní senzory

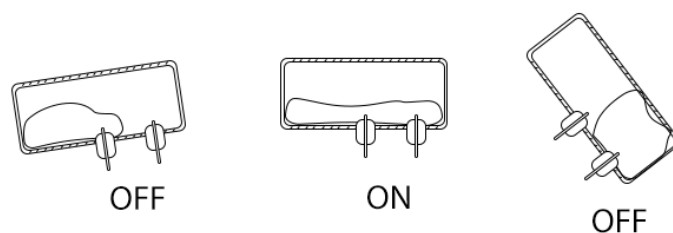
Vlivem neelektrické veličiny mění svou elektrickou vlastnost, odpor, kapacitu, optická nebo indukční. Pro vyhodnocení a přeměnu elektrických vlastností je potřebný zdroj elektrické energie.

7.3 Čidla snímání otřesů

Snímání otřesu je možné provádět různými čidly, níže jsou uvedeny vybrané způsoby detekce otřesů založené na pasivním charakteru součástky.

7.3.1 Rtuťové čidlo

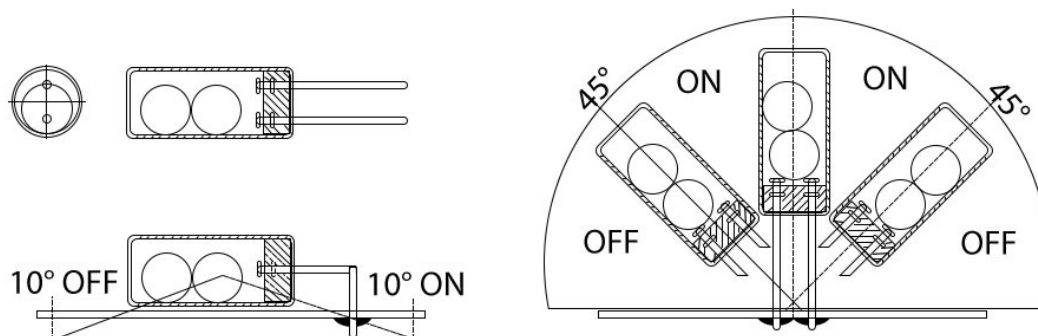
Rtuťová čidla patří do řad jednoduchých čidel náklonu a zrychlení. Používají se tam, kde není potřeba přesného měření a kde postačí signalizace náklonu. Dříve se používaly jako spínače, které měly dobré vlastnosti s velkými proudy okolo 15A, kde nevznikaly přechodové oblouky a byly tedy vhodné do nebezpečných prostředí, kde hrozilo nebezpečí výbuchu. Základní konstrukcí čidla je vyfouklá malá skleněná trubička, ve které je obsažena rtuť v potřebném množství podle velikosti trubičky. Použití skleněného materiálu je zejména z důvodu odolnosti vůči chemickým procesům. V trubičce jsou zakomponovány dva kontakty, které slouží pro indikaci rozepnutého a sepnutého stavu. Při náklonu se rtuť slije do druhého kontaktu a utvoří tak uzavřený obvod. Díky velké měrné hmotnosti a roztažnosti rtuti je náklon sepnutí poměrně malý okolo 3° . Rtuť má stabilní nízký přechodový odpor, neopaluje se a nekoroduje. V současné době se moc rtuťové spínače nepoužívají a nevyrábějí. Důvodem je, vytečení rtuťe při prasknutí skleněného obalu. Rtuť je silně jedovatá, a při odpařování vznikají jedovaté plyny polutanty. Menší čidla jsou odolnější, mají lepší ochranu díky druhému plášti z kovu.



Obrázek 7.1: Olověné čidlo

7.3.2 Kuličkové čidlo

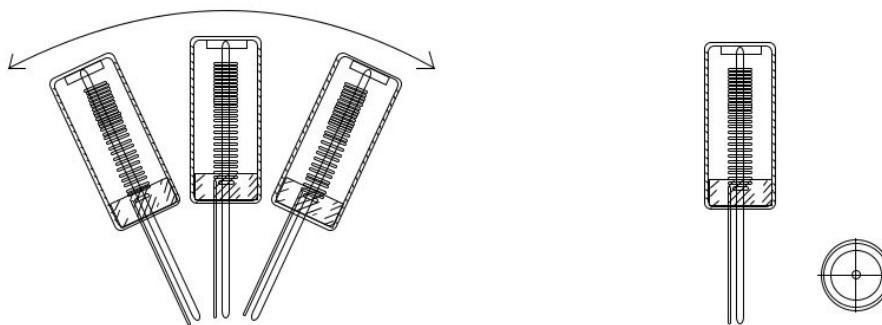
Uvnitř pláště otřesového čidla se nachází dvě kuličky vyrobené z vodivého materiálu, které se v zapnutém stavu (ON) dotýkají kovových nožiček a spojují tak obvod. Tedy čidlo se nachází v klidovém stavu. Při naklonění čidla ve svislé poloze o 15 stupňů, přejde čidlo do stavu (OFF).



Obrázek 7.2: Kuličkové čidlo

7.3.3 Pružinové čidlo

Pod pláštěm čidla se nachází hlavní statická noha, nepohyblivá část. Kolem této nohy je navinutá pružina o průměru materiálu dle takzvané tvrdosti čidla, podle které se určuje citlivost na otřesy. Tato tvrdost čidla se značí v názvu součástky SW - 18010, SW - 18020, SW - 18030. Toto čidlo není závislé na své poloze a je ho tedy možné využít v jakékoli poloze. Samotná funkce probíhá při otřesu, dochází k rozechvění pružiny, která se při určité síle otřesu začne dotýkat statické nohy a dojde tak k propojení mezi kontakty.



Obrázek 7.3: Pružinové čidlo

8 Návrh otřesového čidla

8.1 Mikroprocesor

8.1.1 Požadavky

Jedním z požadavků na mikroprocesor je malá spotřeba elektrické energie, a to z důvodu provozu čidla na bateriovém akumulátoru. Dalším požadavkem je dostatečná kapacita paměti pro nahrání celého programu.

8.1.2 Mikroprocesor Atmega328

Jedná se o obvod založený na architektuře RISC s osmibitovým mikroprocesorem [2]. Použití 8Mhz řídicího krystalu nám umožní zpracovat přibližně 8 milionů instrukcí za vteřinu. Díky nízké pracovní frekvenci zůstává odběr proudu velmi nízký. Obvody mají 32 registrů, které umožňují paralelní zpracování přístupu. Samotnou konstrukci čidla lze zohlednit velikost využití paměti mikroprocesoru, a to tím zda použijeme interní krystal nebo externí. Použitím interního krystalu si zmenšíme plochu čidla, ale zároveň se zmenší kapacita paměti a to z 32 356 bitů na 30 720 bitů téměř o 2kB paměti. Mezi nevýhodu interního krystalu patří jeho malá přesnost. Jedním z požadavků na konstrukci čidla je přesnost taktovací frekvence a to zejména z důvodu využití modulu RFM96W, kde je potřeba generovat přesný signál. Další požadovanou funkcí je takzvaný úsporný režim, kdy dojde na jednotce k poklesu proudu na nejnižší možnou úroveň. ATmega328 [8] obsahuje 5 úsporných módů:

- **Idle Mode** - dojde k zastavení CPU a nechá běžet SRAM, časovače, čítače, USART, 2-wire Serial Interface, SPI port, a interrupt systém.
- **ADC Noise Reduction Mode** - dojde k zastavení CPU a všech I/O modulu, krom asynchronního časovače a ADC, tak aby bylo minimalizováno rušení během A/D převodu.
- **Power-Down Mode** - udržuje obsah registrů, a odpojí oscilátor, to do následujícího interruptu nebo hardwarového resetu zastaví všechny další funkce.
- **Power-Save Mode** - nechá v činnosti asynchronní časovače, zbytek součástky je zastaven.
- **Stand** - v módu je neustále oscilátor v provozu, ostatní části obvodu jsou, uspány. To dovoluje rychlé probuzení s nízkou spotřebou.

8.1.3 Programování mikroprocesoru

Pro zhotovení a nahrání kódu byla zvolena velmi oblíbená a známá platforma Arduino [9], která je vhodná pro rychlý vývoj. Použité programovací prostředí pro platformu je Arduino IDE v. 1.8.0, které je open-source a je možné jej zdarma stáhnout z oficiálních stránek [23]. Podporuje operační systémy Windows, Mac OS X, a Linux. Jako vývojová deska je použita Arduino UNO, s patičí pro mikrokontrolér ATmega328p, který je možné z této patice vyjmout a osadit na svou vlastní desku.

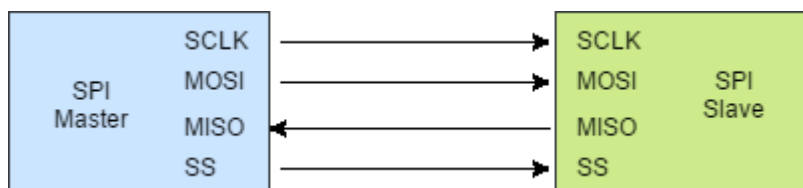
8.2 Návrh plošného spoje, DPS

Návrhem DPS je představeno funkční rozmístění a propojení elektronických součástek. Technika samotné výroby DPS je popsána v této kapitole. Návrh plošného spoje byl vytvořen pomocí editoru plošných spojů EAGLE (Easily Applicable Graphical Layout Editor). Program se skládá ze tří hlavních modulů.

- **Editor spojů** - 16 signálových vrstev, klasické a SMD součástky, rozlišení 1/10.000 mm (0,1 mikronu). Sada knihoven součástek, kterou lze rozšířit o vlastní nebo jiné knihovny. Poměření ploch a meandry pro rychlé obvody.
- **Editor schémat** - Maximálně 99 listů pro jedno schéma, kontrola elektrických pravidel zapojení, prohazování hradel a pinů.
- **Autorouter** - routování až 16 signálových vrstev, strategie propojování nastavitelná pomocí váhových faktorů.

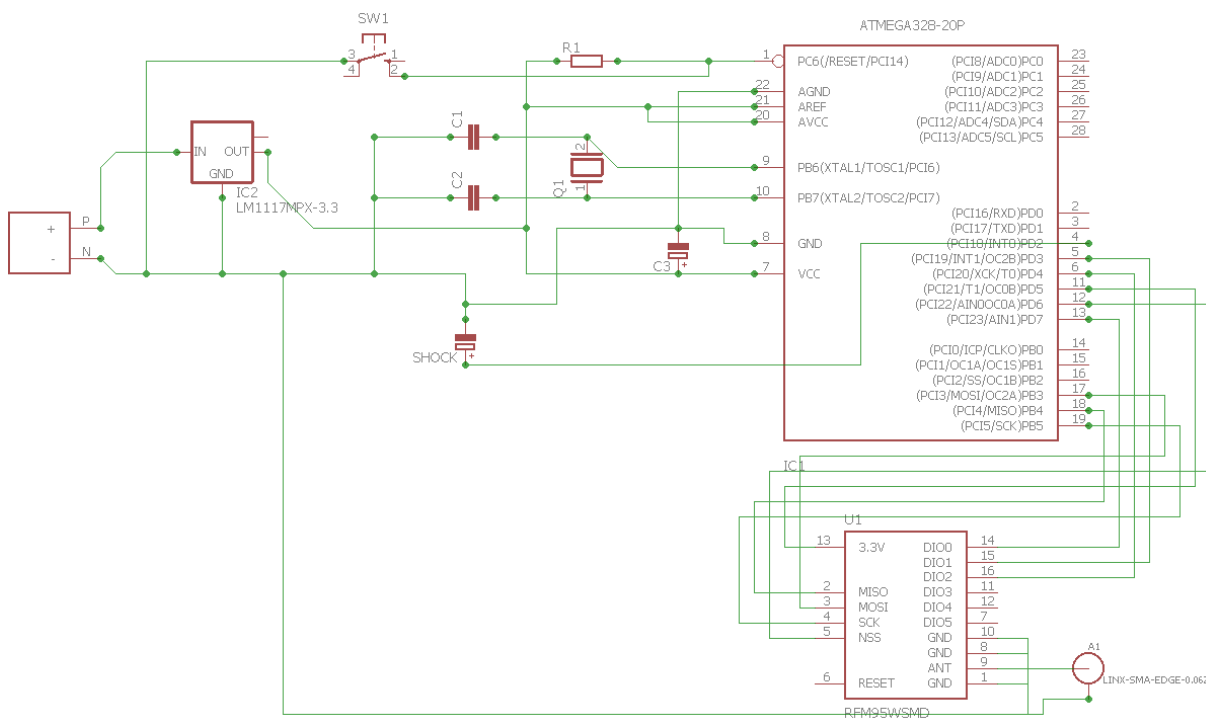
Tyto moduly jsou integrovány do jednoho uživatelského prostředí, a mohou být tak ovládány z jednoho místa. Software je možné použít pod řadou operačních systémů a to Windows, MAC nebo Linux. EAGLE je vydáván ve dvou edicích placené a volně šiřitelné. Volně šiřitelná verze je omezena použitelnou plochou desky, která má maximální velikost v rozsahu 100 x 80 mm (4 x 3.2 palců). Dále mohou být použity maximálně dvě signálové vrstvy plošného spoje a to vrchní a spodní část desky. Na editor schématu je možné využít pouze jediný list na vytvoření schéma. Pro zhotovení návrhu čidla byla freeware verze plně dostačující.

Zapojení součástek je uvedeno níže na obrázku č. 8.1. Z předchozí kapitoly č. 5 byl vybrán prvek pro indikaci otřesů SHOCK a to pružinový snímač. Tato součástka byla zvolena z důvodu svých technických vlastností, kde umístění nemusí být v kolmé poloze a je možné stanovit citlivost snímače pomocí definované tvrdosti pružiny. K napájení desky je použita 12 V baterie. Toto napájení je přivedeno na napětový regulátor od společnosti Texas Instruments s označením LM1117, který snižuje napětovou hladinu na stabilizované napětí 3.3 V, bez nutnosti použití dalších součástek. Toto napětí je důležité pro správný chod celého obvodu. Resetovací obvod slouží k resetování programu implementovaného v mikrokontroléru, díky tomuto spínači s označením ve schématu SW1 je možné provést restart. Dále je v obvodu implementován předřadný rezistor spolu s blokovacím kondenzátorem C3. Pro správný chod obvodu se stará časový generátor signálu takzvaný oscilátor Q1 s kmitočtem o 8 Mhz. Pro stabilizaci signálu jsou k oscilátoru přivedeny blokovací kondenzátory C1 a C2. Komunikačním prvkem v infrastruktuře LoRa byl zvolen z předchozí řešerše modulů uvedené v kapitole č. 3. modul RFM96 od výrobce Hope, v SMD provedení. Pro komunikaci mezi LoRa modulem a MCU jednotkou je zřízena sériová komunikace SPI (Serial Peripheral Interface). Komunikace je realizována pomocí společné sběrnice, samotná adresace je sestavena logickou nulou, kdy je aktivován příjem a vysílání zařízení NSS. Master MCU řídí samotnou komunikaci pomocí hodinového signálu, který je přiveden na vstup SCK modulu.



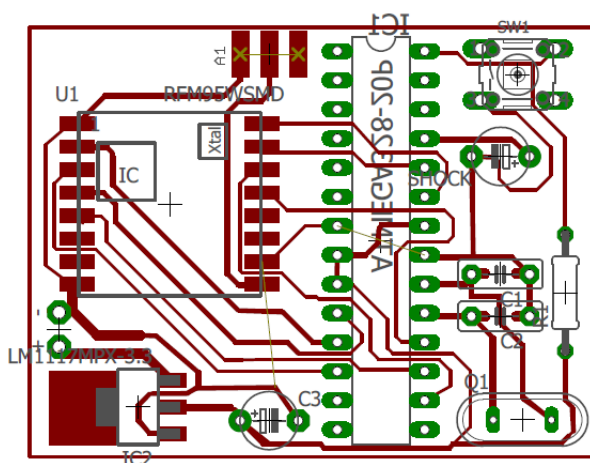
Obrázek 8.1: SPI

Při generování tohoto signálu současně MCU a rádiový modul odesílají data pomocí MOSI (Master Out, Slave In) a vstup MISO (Master In, Slave Out) jak je vyobrazeno na obrázku č. 8.2. Vyvedení antény ze samotného modulu je provedeno konektorem typu SMA do DPS. Po zhotovení návrhu byla vytvořena sestava, pomocí vývojové platformy Arduino. Na této platformě byly jednotlivé součástky rozmístěny na nepájivé pole a otestováno celé kompletní zapojení čidla.



Obrázek 8.2: Blokové schéma čidla

U rozložení součástek vyobrazeném na obrázku č. 8.3 bylo dbáno na minimalizaci proudových smyček a vedení co nejkratších cest v obvodu plošného spoje pro eliminaci rušení. V případě zapojení generátoru časového signálu jsou spoje X1 a X2 vedeny co nejbližěji u sebe, další možností je navázat mezi těmito vodiči zem GND. Obvody s rychlou změnou logických úrovní jsou umístěny v blízkosti vedle sebe pro snížení rušení (DIO0, DIO1, DIO2).



Obrázek 8.3: Deska plošného spoje a rozložení součástek

9 Realizace otřesového čidla

9.1 Struktura kódu

Základní struktura programovacího jazyka Arduino se skládá ze dvou částí `setup`, `loop`. Funkce `setup` je inicializační funkcí a provádí se pouze na začátku kódu. Je využita pro nastavení vstupních a výstupních pinů mikro kontrolérů. Funkce `loop` se provádí neustále dokola. Pro správný chod programu je potřebné tyto dvě funkce vždy použít.

ABP autentizační parametry, pro přihlášení do LoRa WAN sítě, jsou znázorněny v následujícím kódu:

```
const String DEVADDR = "02D1E520";

static const PROGMEM u1_t NWKSKEY[16] = { 0x2A, 0x7E, 0x15, 0x16,
0x28, 0xA5, 0xD2, 0xA7, 0xBA, 0xF7, 0x15, 0x54, 0x59, 0xCF, 0x4F,
0x3C };

static const u1_t PROGMEM APPSKEY[16] = { 0xA4, 0x6E, 0x15, 0x26,
0x28, 0xA3, 0xD2, 0xA6, 0xAB, 0xF7, 0x15, 0x88, 0x09, 0xCF, 0x4F,
0x06 };
```

Pro sestavení LoRa paketu byla využita knihovna *lmic* [30]. V níže uvedeném kódu je namapování vstupních a výstupních pinů z mikrokontroléru vůči frekvenčnímu modulu RFM96W, kde piny pro RX, TX a RESET nebyly využity. NSS je propojeno s PD6 a digitální softwarové vstupy/výstupy s programovatelnými digitálními vstupy na mikrokontroléru (DIO0 na PD7, DIO1 na PD3 a DIO2 na PD4). Nastavení datového toku se vztahuje na generátor impulzů a poskytuje přesné nastavení přenosové rychlosti. Při kontinuálním režimu vysílání musí být datový tok k přenosu přímo vložen do modulátoru pomocí softwarové konfigurace DIO2 (DATA) asynchronním způsobem, není-li použito Gaussovo filtrování, přičemž se DCLK signál nastavený na DIO1 (DCLK) používá k synchronizaci datového toku.

```
const lmic_pinmap lmic_pins = {
    .nss = 6,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = LMIC_UNUSED_PIN,
    .dio = {7, 3, 4},
};
```

V inicializační funkci `setup` byly nastaveny další parametry pro sestavení LoRa paketu a to maximální vysílací výkon TX Power, střední frekvence a Spreading Factor pro nastavení datového toku.

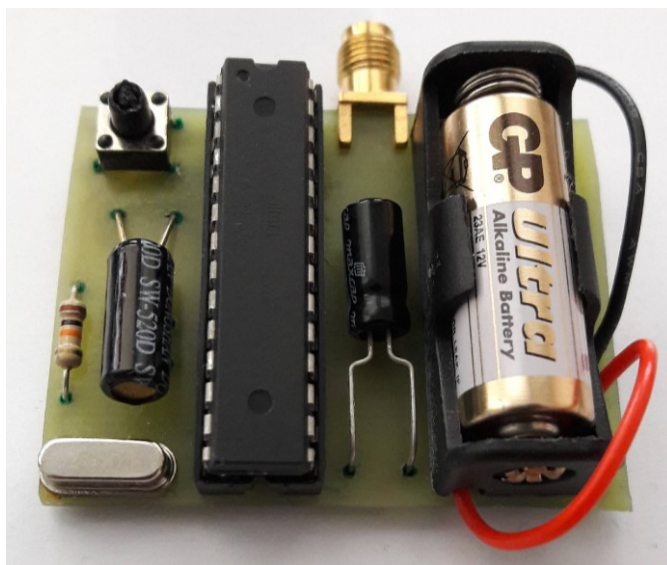
```
void setup() {
    LMIC.freq = 868100000;
    LMIC.txpow = 27;
    LMIC.datarate = DR_SF7; //DR_SF9;
```

```
LMIC.rps = updr2rps(LMIC.datarate);
```

Jelikož se jedná o bateriové čidlo a je předpokládána dlouhá životnost senzoru, je k tomuto účelu zde využita volně šiřitelná knihovna LowPower [27]. Tato knihovna se stará o nastavení mikrokontroléru do různých módů, které snižují spotřebu na minimální úroveň podle zvoleného módu. Tyto úsporné módy jsou popsány v kapitole 8.1.2. Pro čidlo je využit *PowerDown* mód, který udržuje obsah registrů a provádí odpojení oscilátoru. Probuzení čidla je provedeno pomocí takzvaného interruptu, který je nastaven na pin číslo 2, ke kterému je přiveden otřesový snímač. Po probuzení zařízení vyšle zprávu Shock, po odeslání dojde opět k uspaní senzoru. Díky tomuto opatření je možné senzor využívat dlouhodobě, kdy při *PowerDown* stavu je odhadována délka výdrže baterie podle níže uvedeného propočtu na kapacitu baterie 900 mAh / 0.0748 mAh \approx 1,4 Let. Použitá baterie A23L L1022 12 V.

9.2 Výroba plošného spoje, DPS

Pro výrobu plošného spoje byla zvolena takzvaná nažehlovací metoda. Tato metoda je založena na přetisku toneru laserové tiskárny z média na DPS. Jako médium je vhodné použít podklad, na kterém se nám pigment z toneru neuchytí příliš silně. V tomto případě byl použit fotopapír s vysokou lesklostí, na který byl vytištěn motiv plošného spoje. Samotný tisk motivu byl nastaven na nejsytější barvu se zrcadlením, jelikož se motiv po nažehlení zrcadlově obrátí. Dalším krokem byl ořez jednostranného cuprexitu na požadovanou velikost 49mm x 37mm. Následně se cuprexitová deska očistila od hrubých nečistot pomocí jemného smírku o zrnitosti 1000 a umyla mýdlovou vodou, pro zbavení mastnoty a zbylého obroušeného materiálu. Na připravenou desku bylo možné nažehlit samotnou předlohu a přenést tak pigment toneru. Pro odstranění papíru bylo nutné desku ponořit do vody, čímž papír nabobtnal a bylo jej možné jednoduše odstranit z desky, přičemž toner zůstal na DPS. Takto připravená deska byla ponořena do leptacího roztoku chloridu železitého FeCl₃ použitého pro vyleptání obrazce plošného spoje. Délka leptání se odvíjela od vizuální kontroly pomocí prosvícení desky světlem. Před samotným osazením proběhlo vrtání otvorů pro jednotlivé součástky 1 milimetrovým vrtákem a následovala vizuální a detekční kontrola cest pomocí multimetru allosun EM3680. Použitím mikropájky Proskit byly jednotlivé součástky napájeny na desku plošných spojů. Při pájení bylo dbáno na správnou orientaci součástek dle dokumentace. Osazené čidlo je na obrázku č. 9.1.



Obrázek 9.1: Výsledný DPS

Po zapojení obvodu na baterii a našroubování 2dBi pendrekové antény, došlo k oživení senzoru a následně proběhlo ověření, zda senzor odesílá data. Po ověření se kompletní senzor vložil do vyrobené plastové krabičky popsané v kapitole 10. Na závěr se provedlo utěsnění proti vlhkosti pomocí univerzálního silikonu.

10 Návrh krabičky

Realizace čidla a gateway počítá s konstrukcí a realizací krabičky pro ochranu komponentů uvnitř zařízení, vůči vnějším vlivům či poškození samotného polovodiče. Při výběru vhodného 3D modelovacího nástroje se nabízí celá řada programů, bylo přihlášeno k dostupnosti a efektivnosti práce s navrhovaným objektem v modelovacím prostředí. Jako vhodné řešení se jeví webová aplikace TinkerCAD. Výstupem této aplikace je datový formát STL - Standard Triangle Language, tento formát nedokáže vybraná tiskárna PRUSA I3 MK2 zpracovat. Zdrojovým datovým souborem pro 3D tiskárnu je počítačový model výrobku ve formátu G-code. Jedná se o NC - numerical control programovací jazyk, pro řízení automatizovaných přístrojů jako jsou řezné, leštící, svářeční, tiskové či CNC stroje. Pro převod STL souboru do G-code byl použit nástroj Slic3r, kde se také provedlo základní nastavení tiskárny.

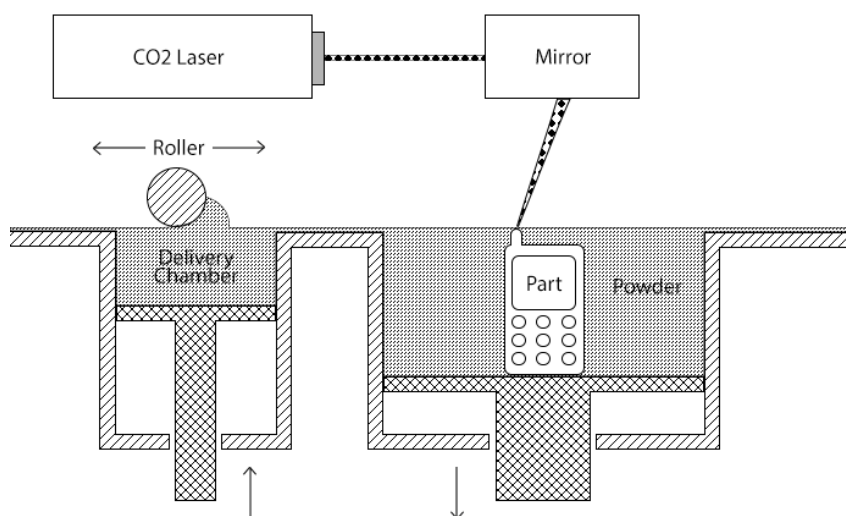
10.1 3D Tiskárny

V této kapitole se zaměříme na základní principy tisku prostorových objektů a jejich metody. Následující rešerše metod nám přiblíží technologický popis tisku na různých typech tiskáren.

10.1.1 Technologické rozdělení

10.1.1.1 *SSL - Selective Laser Sintering*

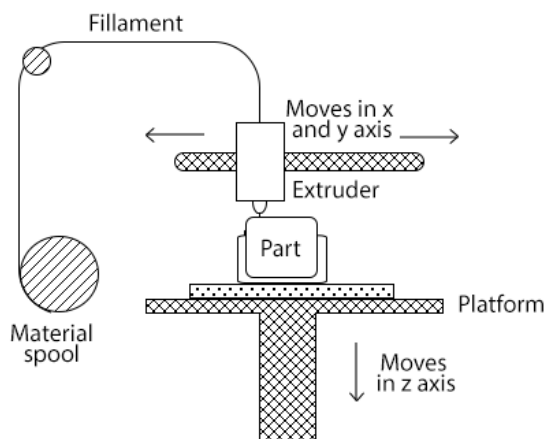
Technologie SSL byla vynalezena v polovině 80 let doktorem Carlem Deckardem. Mezi nejznámější výrobce patří společnost 3D Systems. Jedná se o velmi efektivní metodu, která využívá vysoce výkonný laser. Laserový paprsek dokáže roztavit a zapékat materiál v určité zrnitosti materiálu. Objekt je skládán z jednotlivých vrstev, které jsou následně zapékány. Mezi výhody laserového tisku patří velká škálovatelnost různých takzvaných práškových materiálů. Další výhodou je že není nutné použít přídatné podpěry pro objekt, jelikož je objekt obklopen nespojeným materiálem, který drží formu a tvar objektu. Mezi nevýhody této technologie patří vysoká energetická náročnost a pořizovací náklady.



Obrázek 10.1: *Selective laser sintering*

10.1.1.2 **FDM - Fused Deposition Modeling**

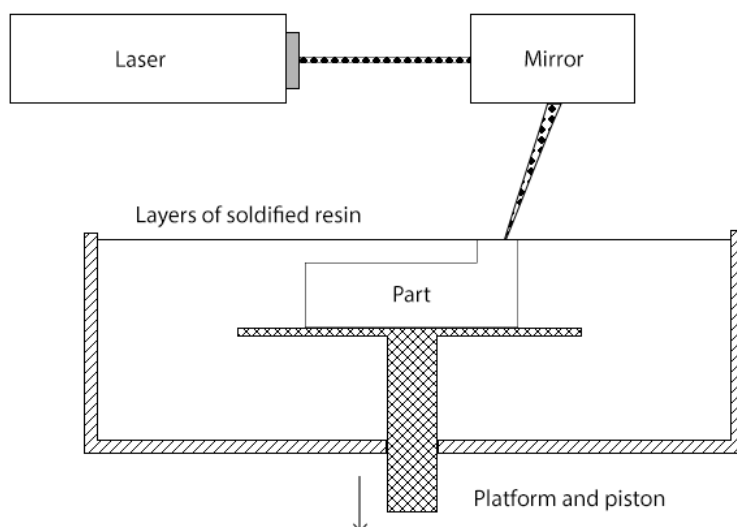
Jedná se o nejpoužívanější metodu 3D tisku, pro výrobu prototypů výrobků. Je zde využit jednoduchý princip tisku, kdy je do tiskové hlavice naváděn materiál pod určitým tlakem. Materiál je následně nanášen po jednotlivých vrstvách na tiskovou plochu. Tato plocha může být nahřívána pro lepší přilnavost materiálů na plotnu.



Obrázek 10.2: *Fused deposition modeling*

10.1.1.3 **SLA - Stereolithography**

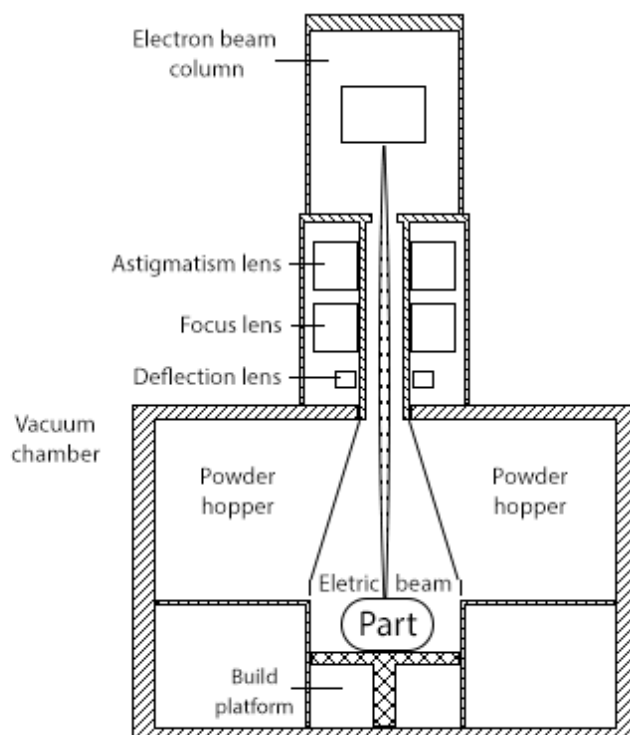
Tato technologii byla patentována Charlesem W.Hullem v roce 1986. Princip této technologie je založen na ultrafialovém laserovém paprsku, který jednotlivé vrstvy vykresluje na hladinu speciální fotocitlivé tekutiny (polymerů). Jakmile je vrstva hotová ponoří se vrstva objektu do kapaliny a proces se opakuje. Nevýhodou této technologie je omezené možnosti z volby výrobního materiálu.



Obrázek 10.3: *Stereolithography*

10.1.1.4 EBM - Electronic Beam Melting

Tisknutí objektů je prováděno pomocí usměrněného proudu elektronů. Tento proud je nastřelován do jemného kovového prachu, který se roztaví a zapeče na daném místě. Materiál kovového prachu se ve většině případů vyrábí z titanové hmoty. Výhodou tohoto materiálu je velká přesnost a technická kvalita výsledného objektu. Nevýhodou této technologie je velká pořizovací cena.



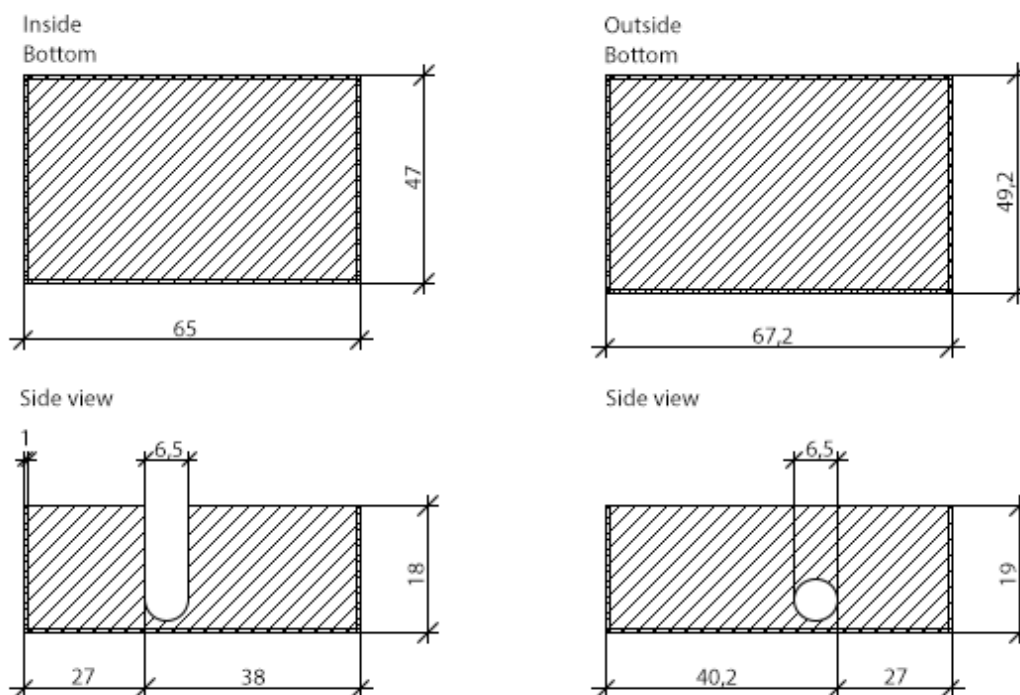
Obrázek 10.4: *Eletronic beam melting*

10.2 Použitá tiskárna

Pro samotný tisk byla zvolena dostupná tiskárna, kterou poskytla katedra telekomunikační techniky. Jedná se o tiskárnu PRUSA I3 MK2 která je založena na FDM - Fused Deposition Modeling tiskové technologii. Základním konstrukčním prvkem je tisková hlavička, do které je naváděn materiál pod určitým tlakem, tento materiál je při průchodu zahříván na tavnou teplotu materiálu a následně vytlačován. Výrobcem této tiskárny je česká společnost PRUSA, která je specializována na výrobu 3D tiskáren, pro hobby použití. Do tiskárny je možné navést různé druhy materiálů ABS, PLA, PET, HIPS, PP a Flex.

10.3 Realizace

Prvním krokem pro vymodelování krabičky, bylo důkladné přenesení rozměrů plošného spoje do 3D modelovacího nástroje. Díky těmto rozměrům byl vytvořen obsah, který je základem vnitřního prostoru krabičky, a následně kolem něj byla vymodelována spodní část. Vnitřní prostor horní části je větší o stěnu spodní a připočítána 0,1 mm mezera, pro těsné lícování krabičky při složení. Samotná tloušťka materiálu je nastavena na 1mm, tato tloušťka je dostatečná pro konstrukční pevnost materiálu, aby odolala mechanickému namáhání a měla dostatečnou pružnost při otřesu.



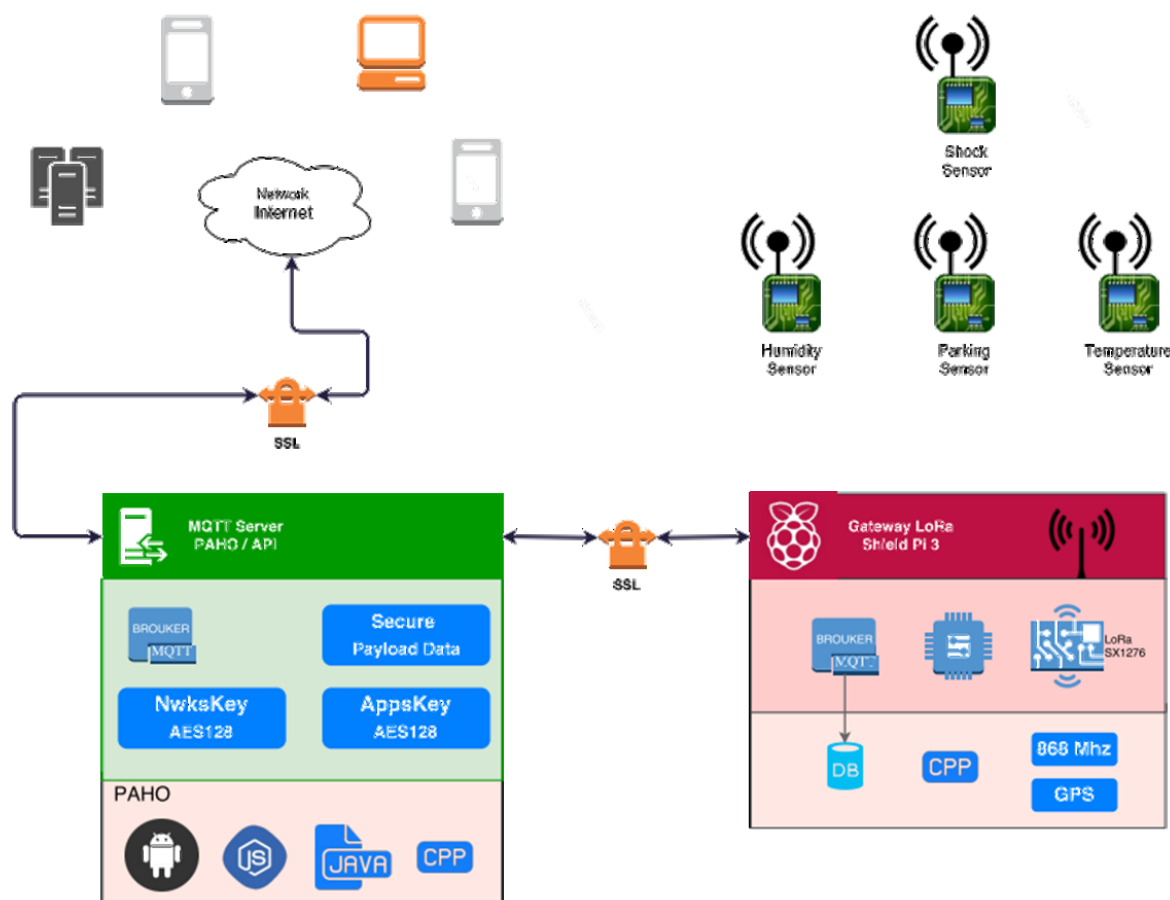
Obrázek 10.5: Návrh krabičky



Obrázek 10.6: Sestavené čidlo spolu s krabičkou

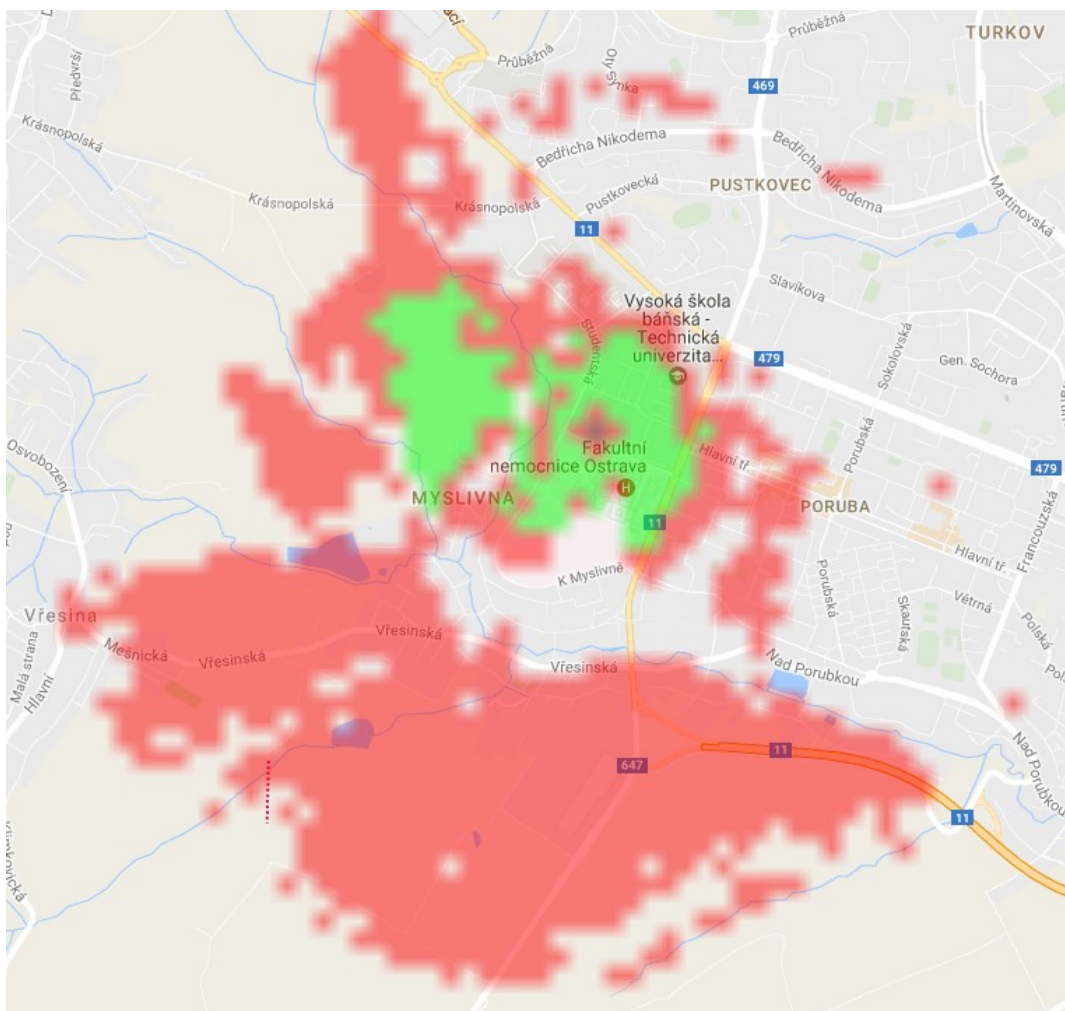
11 VŠB LoRa - Testbed

Součástí této práce byla vytvořena testovací infrastruktura LoRa Testbed, která je součástí kampusu Vysoké školy báňské - Technické univerzity Ostrava, kde je v provozu pro testování, zpracování dat, vývoj čidel a výchozích brán. Tato infrastruktura je založena na technologii LoRa, která je určená pro komunikaci čidel na velké vzdálenosti. Základním prvkem infrastruktury je centrální server, pomocí kterého jsme schopni sesbírat a následně analyzovat nasbíraná data. Tento server je vybaven webovým administračním rozhraním, díky kterému lze do testbedu registrovat další prvky sítě, jak koncové body, tak gateway. Samotná data se po příchodu do sítě v rámci testbedu vyhodnotí a separují od neregistrovaných prvků sítě. Data z neregistrovaných zařízení jsou následně zahozena, nebo logována do souboru pro analýzu. Nedílnou součástí sítě je gateway, kterou jsem vytvořil na základě výsledků provedených rešerší. První testovací brána byla zhotovena pomocí shieldu osazeného rádiovým modulem RFM95W, kde jako MCU jednotkou, bylo napojeno Raspberry Pi. Další testovací výchozí brána byla postavena na koncentrátoru iC880A, který má zhruba pětikrát větší pořizovací cenu než předchozí shield, ale nabízí paralelně přijímat a odesílat pakety až na osmi kanálech. Také nabízí kvalitnější příjem signálu díky již integrovaným filtrům pro odfiltrování nežádoucího pásma. Koncentrátor v propojení s MCU jednotkou Raspberry Pi vytváří bránu, která je srovnatelná s poloprofesionálními bránami, které lze na trhu pořídit. V následujícím schématu je na obrázku č. 11.1 zobrazen koncept komunikace.



Obrázek 11.1: *Koncept komunikace*

V průběhu testování sítě disponujeme desítkami senzorů, které mají integrovaný barometrický modul BMP180 (označením GY-68) od společnosti Bosch, ze kterých v rámci školního kampusu sbíráme testovací data. Čidla jsou rozmístěna v okruhu kampusu na různých sběrných místech (kanceláře, venkovní sloupy osvětlení, atd.). Senzory jsou osazeny různým typem LoRa komunikačního modulu a to zejména RFM95W a MICROCHIP. Nasbíraná data z jednotlivých čidel můžeme analyzovat a graficky zobrazit prostřednictvím aplikace přes webové rozhraní.



Obrázek 11.2: Pokrytí testbedu

Měření pokrytí je provedeno skrze placenou webovou aplikaci Tower Coverage, která je zaměřena na simulaci pokrytí. Tato simulace vyžaduje zadat vstupní parametry pro bránu a senzor. Zadanými parametry jsou výška (budova + stožár) 30 m, výška antény 50 cm, zisk antény 5dBi, zesilovač LoRa modulu +20 dBm. Za pomoci těchto údajů je aplikace schopna vykreslit do Google mapy informace znázorňující pokrytí a to ve dvou stupnicích. Červená jako slabý signál a zelená jako silný, vůči vytvořenému ořesovému čidlu s 2 dBi anténou, od kterého chceme přijímat signál. Mapa pokrytí je uvedena na obrázku č. 11.2.

Závěr

Cílem práce bylo navrhnout a následně prakticky realizovat otřesové čidlo a IoT infrastrukturu dle specifikace LoRa WAN. K realizaci IoT infrastruktury bylo nahlíženo jako komplexní úkol s vytvořením serverové části a vstupní brány, jakožto prvky sítě. Z připravených poznatků a informací o jednotlivých čípech a modulech pro LoRa, bylo možné zvolit vyhovující variantu k sestavení brány a senzoru. Vybraný jednokanálový modul integrovaný do brány je plně dostačující pro testovací účely a chod testbedu. Pro účely širšího použití s větší zátěží je z výsledků doporučen osmi kanálový modul iC880a od společnosti Microchip. Tento modul je také možné propojit s Raspberry Pi pomocí GPIO.

Vlastní přínos práce je v kapitolách 4 až 6, 8 a 9. Ve čtvrté kapitole popisují implementaci IoT GW, která byla realizována s využitím Raspberry Pi spolu s Dragino shieldem, tato nástavba obsahuje čip RFM95W vyráběný společností Hope. Brána byla zhotovena na operačním systému Raspbian, kde je vytvořen program pro sbírání dat, který má implementovanou knihovnu WiringPi a Eclipse IDE. Pátá kapitola představuje vlastní koncept realizace serverové části infrastruktury, kde byl implementován mosquitto broker spolu s ACL kontrolou uživatelů. Dále se zde nachází vlastní koncentrátor dat spolu s webovou aplikací. Závěrem práce je zhotovení otřesového prvku do infrastruktury, kde pomocí navrženého schématu a uvedené metodiky pro výrobu DPS vzniklo čidlo. Tato kompletní struktura skládající se z brány, serveru a čidla byla řádně otestována v reálném provozu a ověřena celková funkčnost. V dnešní době je 3D tisk velmi rozšířený a otevírá tak velké možnosti pro tisk vlastních návrhů, v podobě vlastního přínosu je v deváté kapitole této práce uvedena realizace včetně návrhu krabičky použité pro senzor.

Přínosem této práce, je zhodnocení a otestování dostupných technologií, dále rozšíření možností v oblasti senzorových sítí v kampusu Vysoké školy báňské - Technické univerzity Ostrava a to pomocí vytvořené IoT infrastruktury. Do pilotního využívání je v hlavním univerzitním kampusu v Porubě zapojena především Fakulta elektrotechniky a informatiky a dále mají možnost infrastrukturu využívat i studenti a zaměstnanci Stavební fakulty (FAST), kde je umístěna druhá IoT GW. O rozšíření infrastruktury do kampusu FAST požádali zaměstnanci Katedry dopravního stavitelství, kteří aktivity v oblasti IoT zaregistrovali a pořádali o výrobu specifických čidel pro experimenty v oblasti dopravy. Rovněž můžu zmínit zájem z oblasti průmyslu, např. společnost ELVAC, kde již proběhlo jednání ve věci rozjetí IoT infrastruktury v hlavním areálu společnosti ELVAC v Ostravě Hrabůvce. Můžu konstatovat, že výsledky této diplomové práce již dnes mají své reálné uživatele a další potenciální zájemce o využití, což mne těší. Rovněž bych chtěl zmínit, že vytvořená infrastruktura našla i využití ve výuce, kdy studenti bakalářského studia ve cvičeních realizovali vlastní čidlo, které bylo následně připojeno do LoRa infrastruktury.

Jsem si vědom, že téma Internetu věcí je svým rozsahem obsáhlé, jeho zvládnutí vyžaduje odborné znalosti i zkušenosti, které jsou mnohdy interdisciplinární a má práce je pouze malým střípkem ukazujícím jedno z možných řešení konkrétního zadání. Do budoucna bych se chtěl problematice IoT věnovat i nadále a získané znalosti při řešení této práce dále rozvíjet.

Použitá literatura

- [1] MONK, Simon. Raspberry Pi cookbook. ISBN 978-144-9365-226.
- [2] VÁŇA, Vladimír. Mikrokontroléry ATMEL AVR: Popis procesoru a instrukční soubor. Praha: BEN - technická literatura, 2003. ISBN 80-7300-083-0.
- [3] FAWCETT, Joe, Liam. QUIN a Danny. AYRES. Beginning XML. Fifth edition. ISBN 978-1-118-16213-2.
- [4] HAVERBEKE, Marijn. Eloquent Javascript: a modern introduction to programming. 2nd ed. San Francisco: No Starch Press, c2015. ISBN 978-1-59327-584-6.
- [5] JQuery cookbook. Cambridge: O'Reilly, c2010. ISBN 978-0-596-15977-1.
- [6] OTERO, Cesar a Rob LARSEN. PROFESSIONAL jQuery. ISBN 978-1-118-02668-7.
- [7] GILMORE, W. J. Beginning PHP and MySQL: From Novice to Professional Fourth Edition. Fourth edition. ISBN 978-80-7413-163-9.
- [8] BRTNÍK, Bohumil a David MATOUŠEK. Mikroprocesorová technika: [práce s mikrokontroléry řady ATMEL AVR ATXmega A4]. Praha: BEN - technická literatura, 2011. ISBN 978-80-7300-406-4.
- [9] MONK, Simon. Programming Arduino: getting started with sketches. New York: McGraw-Hill, c2012. ISBN 9780071784221.
- [10] IYENGAR, S. S. Fundamentals of sensor network programming: applications and technology. Hoboken, N.J.: John Wiley, c2011. ISBN 978-0-470-87614-5.
- [11] SIGFOX: The Global Communications Service Provider for the Internet of Things (IoT) [online]. Labège: SIGFOX, c2016 [cit. 2016-04-24]. Dostupné z: <<https://www.sigfox.com>>
- [12] Cloud Customer Architecture for IoT. Cloud Standards Customer Council [online]. 2016. Dostupné z: <<http://www.cloud-council.org/deliverables/CSCC-Cloud-Customer-Architecture-for-IoT.pdf>>
- [13] The Internet of Things: An Overview. The Internet Society (ISOC) [online]. 2015 [cit. 2015-10]. Dostupné z: <https://www.internetsociety.org/sites/default/files/ISOC-IoT-Overview-20151014_0.pdf>
- [14] Internet of Things: Wireless Sensor Networks. International Electrotechnical Commission (IEC) [online]. 2014 [cit. 2014-11]. Dostupné z <<http://www.iec.ch/whitepaper/pdf/iecWP-internetofthings-LR-en.pdf>>
- [15] KOFLER, Michael a Bernd ÖGGL. PHP 5 a MySQL 5: průvodce webového programátora. Brno: Computer Press, 2007. ISBN 978-80-251-1813-9.

- [16] LoRa/GPS HAT Documentation. Dragino [online]. 2017 [cit. 2017-]Last modified on 24 March 2017,
- [17] RFM95/96/97/98(W) - Low Power Long Range Transceiver Module. hoperf [online]. 2006 [cit. 2015-05-23]. Dostupné z: <http://www.hoperf.com/upload/rf/RFM95_96_97_98W.pdf>
- [18] EADY, Fred. Hands-on ZigBee: implementing 802.15.4 with microcontrollers. Boston: Elsevier/Newnes, c2007. Embedded technology series.
- [19] LAMPKIN, Valerie. Building smarter planet solutions with MQTT and IBM WebSphere MQ telemetry. ISBN 0738437085.
- [20] ISO/IEC 20922:2016. Internal Organization for Standardization (ISO) [online]. 2016 [cit. 2016-06]. Dostupného z: <<https://www.iso.org/standard/69466.html>>
- [21] IEEE 802.15.4. IEEE Standards Association [online]. 2011 [cit. 2011-04-15]. Dostupné z: <http://ecee.colorado.edu/~liue/teaching/comm_standards/2015S_zigbee/802.15.4-2011.pdf>
- [22] The Internet of Things: Sizing up the opportunity. McKinsey [online]. 2014 [cit. 2014-12]. Dostupné z: <<http://www.mckinsey.com/industries/semiconductors/our-insights/the-internet-of-things-sizing-up-the-opportunity>>
- [23] Arduino IDE, ARDUINO 1.6.9. Arduino [online]. 2016 [cit. 2016-05-10]. Dostupné z: <<https://www.arduino.cc/en/Main/Software>>
- [24] Frequency Modulation (FM) Synthesis. Tamara Smyth, School of Computing Science, Simon Fraser University [online]. 2013 [cit. 2013-10-16]. Dostupné z: <https://www.cs.sfu.ca/~tamaras/freqMod/freqMod_4up.pdf>
- [25] Company Overview. Semtech [online]. 2017 Dostupné z: <<http://www.semtech.com/company/>>
- [26] OpenWrt [online]. 2004 [cit. 2004-03-05]. Dostupné z: <<https://github.com/openwrt>>
- [27] Low-Power. Rocketscream [online]. 2016 [2016-04-01]. Dostupného z: <<https://github.com/rocketscream/Low-Power>>
- [28] RDONS PROJECTS. WiringPi [online]. 2012 [cit. 2012-12-13]. Dostupné z: <<https://projects.drogon.net/raspberry-pi/wiringpi/>>.
- [29] GNU Lesser General Public License. GNU Operating System [online]. 2007 [2007-06-29] Dostupného z: <<http://www.gnu.org/copyleft/lesser.html>>
- [30] Arduino-limc. Matthijs Kooijman [online]. 2016 [cit. 2016-08-09] Dostupného z: <<https://github.com/matthijskooijman/arduino-lmic>>

- [31] ROSEBROCK, Eric a Eric FILSON. Linux, Apache, MySQL a PHP: instalace a konfigurace prostředí pro pokročilé webové aplikace. Praha: Grada, 2005. Průvodce (Grada). ISBN 80-247-1260-1.
- [32] Operační systém. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-04-27]. Dostupné z: <https://cs.wikipedia.org/wiki/Opera%C4%8Dn%C3%AD_syst%C3%A9m>
- [33] LoRa RF Bands | LoRa Frequency Bands | RF Wireless World. In: RF Wireless World [online]. San Francisco (CA): Wireless Vendors and Resources, 2012 [cit. 2017-04-27]. Dostupné z: <<http://www.rfwireless-world.com/Tutorials/LoRa-frequency-bands.html>>
- [34] SB Admin 2. In: Start Bootstrap [online]. San Francisco (CA): David Miller at Blackrock Digital, 2009 [cit. 2017-04-27]. Dostupné z: <<https://startbootstrap.com/template-overviews/sb-admin-2>>

A Příloha

Obsahem přiložené elektronické verze diplomové práce jsou veškeré zdrojové kódy pro realizaci LoRa infrastruktury včetně datasheetů a informací k hardwarovým komponentům. Dále jsou zde uvedeny PCB návrhy a fotogalerie postupu práce.